

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

“MARCADOR ELECTRÓNICO DE BASKET”

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN
ELECTRÓNICA Y TELECOMUNICACIONES**

MEJÍA CHILUISA CARLOS ROBERTO
carlosmejiachiluisa@yahoo.com

DIRECTOR: ING. MÓNICA VINUEZA
monica.vinueza@epn.edu.ec

QUITO, MARZO 2008

DECLARACIÓN

Carlos Roberto Mejía Chiluisa, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado por ningún grado o calificación profesional, y que todo el contenido de este trabajo ha sido consultado en las referencias bibliográficas que se incluyen en este documento.

Por medio de la presente declaración, cedo mis derechos de propiedad intelectual correspondientes a este trabajo a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normativa institucional vigente.

Carlos Roberto Mejía Chiluisa

CERTIFICACIÓN

Certifico que este trabajo fue desarrollado por Carlos Roberto Mejía Chiluisa, bajo mi supervisión.

Ing. Mónica Vinueza

DIRECTORA DEL PROYECTO

AGRADECIMIENTOS

Agradezco a Dios que me ha otorgado el don del conocimiento, a mis padres por su grandioso esfuerzo, a la Ing. Mónica Vinueza por su guía como directora de este proyecto; a mis compañeros que me apoyaron para la realización de este.

Este trabajo también lo dedico a las personas que quiero y aprecio.

Carlos Roberto Mejía Chiluisa

DEDICATORIA

Dedico este trabajo a mis padres, que con infinito amor y sabiduría han sabido guiar mi vida por el sendero de la justicia y la libertad, a fin de engrandecer cada día más a mi patria.

Carlos Roberto Mejía Chiluisa

RESUMEN

En el Capítulo I se describe al microcontrolador PIC16F628A, su arquitectura y características eléctricas; para luego proceder a dar a conocer los displays de 9*12cm que se utilizó porque ofrecen una mejor visualización.

Es así que para manejar dichos displays se da a conocer los decodificadores 7447 y para multiplexar o activar todos los displays a la vez se indican las características de los transistores bipolares NPN y PNP, así como también los transistores darlington que permiten manejar de una mejor manera los displays para una mejor visualización.

En el Capítulo II se detalla el diseño y construcción del equipo, empezando por identificar las etapas del circuito, las cuales son el cronómetro y el contador de aros del equipo A y B. Luego se procede a diseñar dichas etapas, esto consiste en realizar los diagramas esquemáticos; así como también los circuitos impresos respectivos. Continuando con la programación del microcontrolador, en esta parte se da a conocer el software que existe para compilar y grabar los programas hacia el PIC de una manera fácil y rápida. Finalmente se construye el equipo, es decir, se realiza el montaje de todos los elementos que conforman el circuito; para luego ensamblarlo en forma correcta y segura. Seguidamente se realiza pruebas en lugares abiertos y cerrados para observar su funcionamiento; como existen algunos defectos se realiza los ajustes respectivos. Esto se concluye con un balance del costo del equipo.

En el Capítulo III se mencionan las respectivas Conclusiones y Recomendaciones del proyecto, se completa determinada información en los anexos; también se elabora un manual de mantenimiento del equipo y finalmente se incluye la Bibliografía utilizada.

PRESENTACIÓN

El PIC16F628A es de fácil manejo, se le puede programar fácilmente sus memorias sean estas: la memoria de programa, EEPROM y la RAM; como sus puertos el A y B así como también sus dos comparadores y lo más importante que posee un oscilador interno de 4Mhz que es muy útil para realizar proyectos que no requieren demasiada precisión. Por otra parte este PIC puede funcionar con un voltaje de 3V hasta 5.5V y entregar por cada puerto un total de corriente de 200mA, es decir que cada línea entrega 25mA, lo cual es suficiente para el control de dispositivos tal como, transistores, circuitos integrados u opto acopladores.

Este microcontrolador se lo puede aplicar para el control de motores de corriente continua o corriente alterna, motores paso-paso, servomotores, manejo de módulos LCD, manejo de luces, comunicación serial y en este caso se ha utilizado este microcontrolador para el manejo de displays de 7 segmentos de ánodo común, es así que se logra conformar un marcador electrónico de basket, el cual será muy útil en coliseos o a su vez en canchas deportivas con su respectiva adecuación para su correcta visualización.

Este proyecto es aplicable para ser visualizado en coliseos cerrados para observar el tiempo y marcadores de los equipos en un partido de basket, pero si se desea darle otra aplicación con una adecuada variación en la programación de los PICs y el circuito de control sería muy útil, en instituciones bancarias en el caso de que se requiera un control de turnos, en empresas o instituciones públicas donde se requiera un reloj o también en el caso de que se requiera un marcador para fútbol.

Se recomienda que esta tesis sea investigada o analizada por compañeros estudiantes, para una mejora en su visualización en espacios abiertos como en canchas deportivas de basket; así como también en el circuito de control ya que este proyecto involucra para el control a distancia varios metros de cable UTP; se ha investigado que el control se lo puede realizar por radiofrecuencia sin cables.

ÍNDICE

CAPÍTULO I

FUNDAMENTOS TEÓRICOS

1.1. ARQUITECTURA DEL PIC16F628A.....	1
1.1.1. Qué es el PIC.....	1
1.1.2. Estructura del PIC.....	3
1.1.3. Tamaños de memoria.....	7
1.1.4. Los puertos del PIC.....	9
1.1.5. Set de instrucciones del PIC 16F628A.....	12
1.1.5.1. Lista de todos los comandos Basic del Compilador.....	12
1.1.5.2. Utilización del set de instrucciones.....	13
1.1.6. Grabadores de PICs.....	27
1.2. TIPOS DE DISPLAYS 9*12 CM.....	33
1.2.1. Display de 7 segmentos LMS-40101BS.....	35
1.2.2. Características Eléctricas del display LMS-40101BS.....	36
1.3. MANEJO DE DECODIFICADORES.....	38
1.3.1. Decodificador 7442.....	39
1.3.2. Decodificador 7447.....	39
1.4. TRANSISTORES BIPOLARES, DARLINGTON.....	40
1.4.1. Regiones operativas del transistor.....	42
1.4.2. Transistor NPN 2N3904 y PNP 2N3906.....	42
1.4.3. Transistor Darlington.....	44
1.4.4. Transistores Darlington TIP127 y TIP122.....	46
1.4.5. Transistores Darlington ECG 268.....	47

CAPÍTULO II

DISEÑO Y CONSTRUCCIÓN DEL EQUIPO

2.1. IDENTIFICACIÓN DE LAS ETAPAS DEL CIRCUITO.....	48
2.2. DISEÑO DE LAS ETAPAS.....	50
2.2.1. Cronómetro.....	50
2.2.1.1. Funcionamiento del Cronómetro.....	52
2.2.2. Contador de aros del Equipo A/B.....	54
2.2.2.1. Funcionamiento del Contador de aros del Equipo A/B.....	55
2.2.3. Circuito Total.....	57
2.3. PROGRAMACIÓN DEL MICROCONTROLADOR.....	58
2.3.1. Diagrama de Flujo del Cronómetro.....	60
2.3.2. Explicación del programa del Cronómetro.....	61
2.3.3. Programa del Cronómetro.....	61
2.3.4. Diagrama de flujo del Contador de aros del Equipo A/B.....	69
2.3.5. Explicación del programa del Contador de aros del Equipo A/B.....	70
2.3.6. Programa del Contador de aros del Equipo A/B.....	71
2.4. CONSTRUCCIÓN DEL EQUIPO.....	75
2.4.1. Tarjetas de circuito impreso.....	75
2.4.2. Montaje de los elementos.....	76
2.4.3. Montaje y conexión de las tarjetas.....	77
2.4.4. Ensamblaje del Marcador Electrónico.....	77
2.5. PRUEBAS Y AJUSTES.....	79
2.5.1. Pruebas.....	79
2.5.2. Ajustes.....	79
2.6. COSTO DEL EQUIPO.....	80

CAPÍTULO III

CONCLUSIONES Y RECOMENDACIONES

3.1. CONCLUSIONES.....81

3.2. RECOMENDACIONES.....83

BIBLIOGRAFÍA.....84

ANEXOS

ANEXO A

Características de los Marcadores de venta.....A1-A2

ANEXO B

Circuitos Impresos del proyecto.....A3-A6

ANEXO C

Manual del Equipo.....A7-A11

ANEXO D

Manual del usuario.....A12-A15

ANEXO E

Fuente switching.....A16-A17

ANEXO F

Cableado UTPA18-A19

ÍNDICE DE FIGURAS

Figura 1.1.....	2
Figura 1.2.....	3
Figura 1.3.....	4
Figura 1.4.....	7
Figura 1.5.....	8
Figura 1.6.....	8
Figura 1.7.....	9
Figura 1.8.....	10
Figura 1.9.....	11
Figura 1.10.....	12
Figura 1.11.....	13
Figura 1.12.....	30
Figura 1.13.....	31
Figura 1.14.....	32
Figura 1.15.....	33
Figura 1.16.....	33
Figura 1.17.....	34
Figura 1.18.....	35
Figura 1.19.....	35
Figura 1.20.....	36
Figura 1.21.....	37
Figura 1.22.....	38
Figura 1.23.....	38
Figura 1.24.....	39
Figura 1.25.....	40
Figura 1.26.....	40
Figura 1.27.....	41
Figura 1.28.....	43
Figura 1.29.....	44
Figura 1.30.....	46
Figura 1.31.....	47

Figura 2.1.....	48
Figura 2.2.....	49
Figura 2.3.....	51
Figura 2.4.....	52
Figura 2.5.....	54
Figura 2.6.....	55
Figura 2.7.....	56
Figura 2.8.....	57
Figura 2.9.....	58
Figura 2.10.....	59
Figura 2.11.....	59
Figura 2.12.....	60
Figura 2.13.....	68
Figura 2.14.....	69
Figura 2.15.....	76
Figura 2.16.....	76
Figura 2.17.....	77
Figura 2.18.....	78
Figura 2.19.....	78
Figura 2.20.....	80



CAPÍTULO I

1. FUNDAMENTOS TEÓRICOS

1.1. ARQUITECTURA DEL PIC16F628A

1.1.1. ¿QUÉ ES EL PIC? ⁴

Los circuitos integrados programables (Programmable Integrated Circuits = PIC) son componentes sumamente útiles en la Electrónica de Consumo. Aún cuando son conocidos desde hace más de veinte años, existen en la actualidad nuevos tipos que cumplen con una serie de requisitos y características sumamente útiles.

Como una primera aproximación se puede definir a un PIC como “un chip que me permite obtener un circuito integrado a la medida”, es decir se puede hacer que el PIC se comporte como un procesador de luminancia o un temporizador o cualquier otro sistema mediante un programa que se graba en una memoria ROM interna.

Los microcontroladores PIC son en el fondo procesadores similares a otros tipos, como por ejemplo la familia de los microprocesadores X86, 80486, Pentium y muchos otros que usan una arquitectura interna del tipo Von Neumann.

En este tipo de arquitectura los datos y la memoria del programa se encuentran en el mismo espacio de direcciones. En realidad un microprocesador y un microcontrolador no son la misma cosa. Los PICs son microcontroladores, es decir, una unidad que posee en su interior al microprocesador y a los elementos indispensables para que pueda funcionar como una minicomputadora en un solo chip.

⁴ Reyes, Carlos. (2004). “Aprenda rápidamente a programar microcontroladores PIC”, primera edición, Ecuador.

Un microprocesador es solamente la unidad central de procesos o CPU, la memoria, los puertos y todos los demás periféricos son exteriores. La programación de un microprocesador es, por lo tanto, una tarea compleja porque deben controlarse todos estos dispositivos externos.

Un microcontrolador integra la CPU y todos los periféricos en un mismo chip. El programador se desentiende de una gran cantidad de dispositivos y se concentra en el programa de trabajo. Esta circunstancia da lugar a una gran pérdida de tiempo porque los datos tienen que ser retirados de la memoria y llevados a la CPU (Central Processor Unit) y viceversa. Esto significa que la computadora dedica la mayor parte del tiempo al transporte de datos de ida o de vuelta, en lugar de usar este tiempo para trabajar sobre los datos.

Los PICs emplean un conjunto de instrucciones del tipo RISC (Reduced Instruction Set Computer). Con el RISC se suele ejecutar la mayoría de las instrucciones con un solo pulso del clock. Con las instrucciones que se usan en otros equipos del tipo CISC (Complex Instruction Set Computer), se logran instrucciones más poderosas, pero a costa de varios ciclos del clock. En el bien conocido procesador 68HC11 de Motorola se requieren a veces hasta 5 ciclos del clock para ejecutar una instrucción.



Figura 1.1. Arquitectura simplificada del PIC 16F628A ¹

¹ Corrales, Santiago. (2005), " *Electrónica práctica con microcontroladores PIC*", primera edición, Ecuador.

1.1.2. ESTRUCTURA DEL PIC ¹

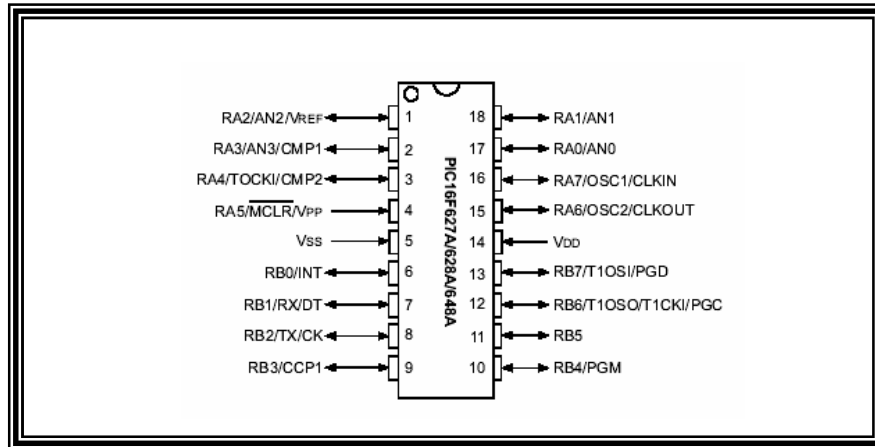


Figura 1.2. Estructura de un PIC ¹

Los microcontroladores poseen una memoria de programa que es el lugar donde deben alojarse los datos que le indiquen al chip qué es lo que debe hacer; una memoria de datos donde ingresen las señales que debe procesar el programa, una unidad aritmética y lógica donde se desarrollen todas las tareas, una unidad de control que se encargue de supervisar todos los procesos y puertos de entrada y salida para que el PIC tenga contacto con el exterior.

Un microcontrolador como cualquier circuito integrado analógico tiene entradas, salidas y algunos componentes exteriores necesarios para procesar las señales de entrada y convertirlas en las señales de salida (Figura 1.2.).

El 16F628A, requiere por supuesto una tensión de fuente de 5V (VDD) aplicada con respecto al terminal de masa (VSS). Posee dos puertos de salida, el A y el B, cuyos terminales son marcados RA0 al RA4 y RB0 al RB7. Estos puertos pueden ser programados como de entrada o de salida. El terminal 4 opera como reset pero también cumple funciones de carga de memoria de programa cuando es excitado con pulsos de 15V. El terminal RA4 (pin 3) también tiene funciones como entrada de un temporizador y RBO (pin 6) cumple también funciones como entrada de interrupción. Ahora bien, la

mayoría de los microcontroladores (sean de Microchip, o de National, Motorola, Philips, etc.), se comportan de forma similar, el PIC16F628A cuya arquitectura se puede observar en diagrama de bloques en la (Figura 1.3.).

Se debe observar primero los bloques externos. Existe un cristal que se conecta en OSC1 y OSC2 para generar el CLOCK del sistema. Luego una señal de entrada llamada MCLR negada, y también para el RESET (es por eso que tiene un doble uso) y, por último, dos puertos paralelos de I/O (entrada o salida) llamados puerto A y puerto B. Uno de las pines del puerto A puede ser utilizado como entrada de interrupciones (este pin especial hace que el microprocesador deje de realizar la tarea que estaba ejecutando y pase a realizar otra tarea alternativa; cuando la termina vuelve a su programa original).

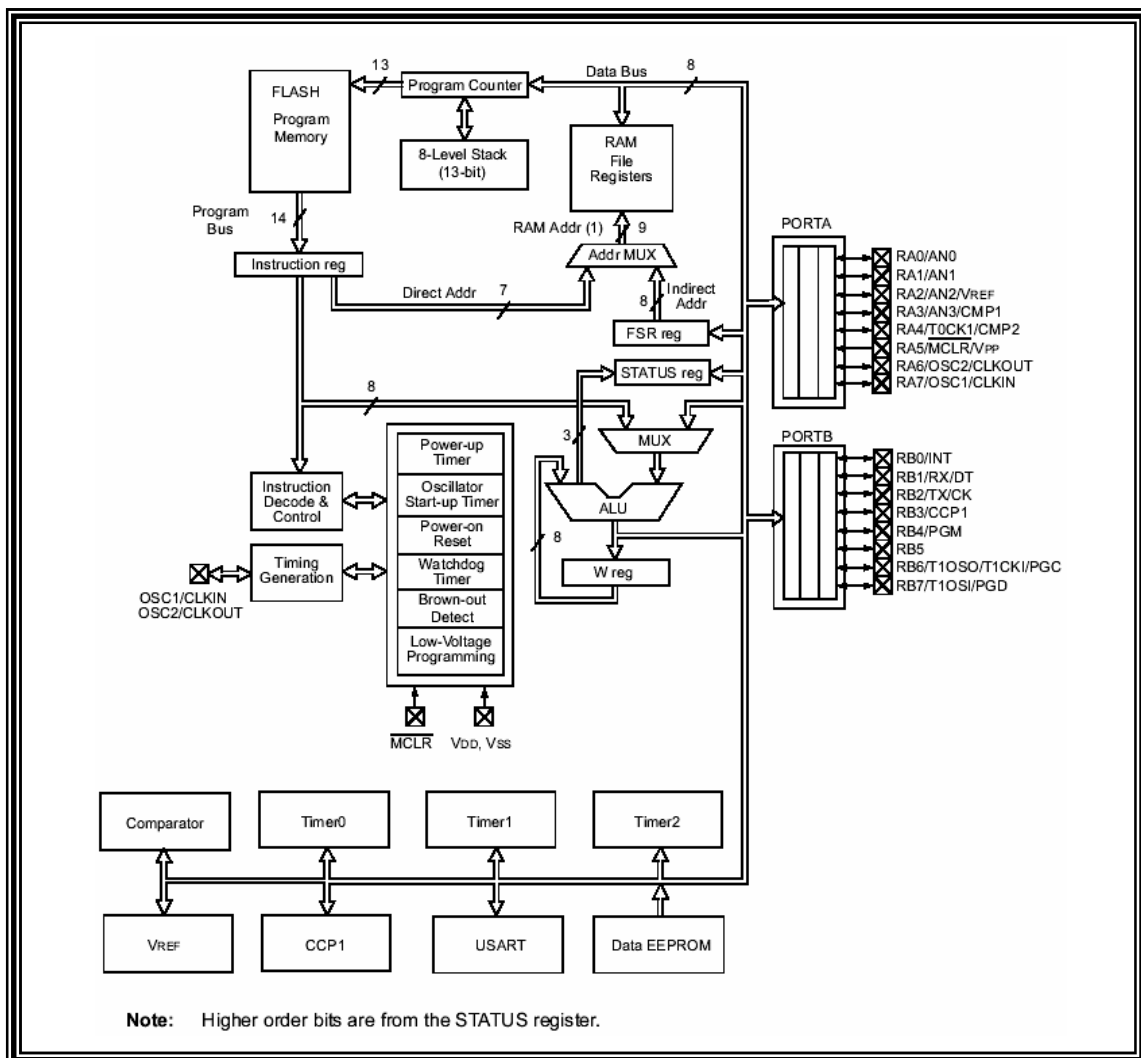


Figura 1.3. Diagrama de bloques del PIC⁵

⁵ <http://www.microchip.16F628A.pdf.com>

El bloque más grande, que es el de temporizadores está dedicado a mejorar el funcionamiento; pero sin influir directamente en el flujo de señales, se ve un temporizador de encendido, un temporizador de arranque del oscilador de CLOCK, un circuito de reset y un circuito llamado de vigilancia o WATCHDOG, los dos primeros bloques procuran un arranque ordenado para no producir una carga al mismo tiempo sobre la fuente.

Por último, existe un circuito con el nombre de “perro guardián”. Su función es estar vigilante el máximo de tiempo que tarda el microprocesador en completar su programa y en caso de superarse ese tiempo, provocar un reset automático porque el microprocesador se quedó trabado en alguna parte de su programa.

Este bloque de circuitos no trabaja independientemente sino que requiere conexiones al exterior y al interior del dispositivo. Por ejemplo, no siempre son utilizados y es el programa quien determina su utilización y además ajusta sus parámetros. Esto se realiza a través del bloque de control o decodificador de instrucciones.

La sección de arriba a la izquierda en donde se ve la memoria de programa, el contador de programa, el registro de instrucciones y la pila o STACK de 8 niveles. Cuando se habla de registros se refiere a pequeñas unidades de memoria transitoria, construida por lo general con un registro de desplazamiento, son memorias volátiles que se utilizan para guardar información por un tiempo mínimo; con el fin de realizar una operación compleja de varios pasos.

El contador de programa es el responsable de que el microprocesador vaya analizando las instrucciones en orden ascendente. Este guarda el número de instrucción en el STACK y la instrucción misma la pasa al registro de instrucciones desde donde se envía al resto del microprocesador. El STACK

es, en realidad, una pila de registros (en el ejemplo utilizado hay 8), debido a que el programa puede tener derivaciones.

Cuando se termina de ejecutar un loop (lazo del programa) se debe volver al mismo punto del programa en donde se había producido la división del mismo y eso es posible porque ese número de instrucción quedó guardado en uno de los registros de la pila. Es común que un loop tenga, a su vez, un loop secundario y cuando se ejecuta ese loop secundario se debe volver al mismo punto del loop primario, eso se consigue con guardar ese número de instrucción del loop secundario en otro registro de la pila.

Los bloques responsables de efectuar operaciones matemáticas y lógicas binarias es la ALU; el nombre ALU proviene de Arithmetic Logic Unit (unidad aritmética y lógica). En este sector es imprescindible utilizar un registro ya que una operación aritmética o lógica siempre se efectúa entre dos números. Los números binarios que deben procesarse se toman de la memoria de datos, el primero se acumula en el registro de trabajo o registro W (de Work = trabajo) el segundo es el presente en el instante en que se llama a la memoria de datos. Como las operaciones pueden ser encadenadas (cuando el resultado sirve como operando de la siguiente operación, tal como el caso de un producto) el registro W tiene un retorno a la ALU.

La ALU está comandada por el bloque MUX (Multiplexador). En efecto, la ALU requiere que se le envíen números para procesar que le lleguen desde la memoria de datos, pero antes se la debe predisponer para que efectúe la operación requerida (comparación, rotación de dígitos, etc.).

El registro de estado o estatus colabora durante las operaciones matemáticas, para realizar una resta: primero ubica el primer número, luego el segundo y después comienza a analizar los bits menos significativos (las unidades), pero si el número de arriba es menor que el número de abajo, entonces toma prestado de la columna de las decenas, luego debe recordar esto porque el número de arriba en la columna de las decenas se redujo en una unidad. En realidad, aunque se trate de una operación entre dos números su ejecución

requiere guardar lo que se llama acarreo en otro registro y éste no es otra cosa más que el registro STATUS. El PIC16F628A contiene además de todo lo visto una memoria RAM de registros que puede ser llamada desde el registro de instrucción a través de un multiplexador de direcciones.

Ahora bien, un microcontrolador sin programa no sabe hacer nada, se le debe enseñar a incrementar o reducirse a intervalos regulares, a encender luces, a sumar, a restar, etc.

Esto significa programarlo y eso se realiza con una plaqueta de programación que depende de cada marca y modelo de microcontrolador, antiguamente los microprocesadores tenían una ventanita transparente y era necesario exponerlos a la luz ultravioleta para borrar su memoria e introducir un programa nuevo. Actualmente cuentan con memorias que no tienen este requisito, basta con cargarlos con un programa para que se borre el programa anterior.

1.1.3. TAMAÑOS DE MEMORIA ¹⁵

MEMORIA DEL PIC 16F628A

MEMORIA DE PROGRAMA FLASH(KBYTES)	MEMORA DE DATOS RAM (BYTES)	MEMORIA DE DATOS EEPROM (BYTES)
2048x14	224x8	128x8

Figura 1.4. Tamaños de memoria ¹⁵

Organización de la Memoria de Programa: El PIC16F628A posee 2Kx14 de espacio en la memoria de programa, que va desde la dirección (0000h-07FFh); el vector de reset esta localizado en la dirección 0000h y el vector de

interrupción en la dirección 0004h, la memoria de datos es de 224 bytes; además una memoria EEPROM de 128 bytes. De estas memorias la más manipulable y programable es la memoria de programa, la RAM y la EEPROM requieren de instrucciones especiales para acceder a su programación y borrado.

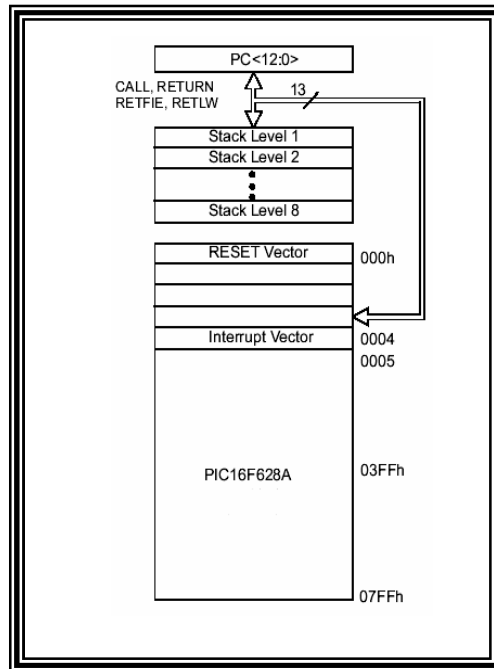


Figura 1.5. Organización de la memoria de programa ⁵

Organización de la memoria de datos: La memoria de datos esta dividida en 4 bancos, los cuales contienen los registros de propósito general y los registros de funciones especiales (SFR). Los registros de funciones especiales están localizados en las primeras 32 direcciones de cada banco, a continuación se muestra la distribución de los registros de propósito general en cada banco.

	PIC16F628A
Bank0	20-7Fh
Bank1	A0h-FF
Bank2	120h-14Fh, 170h-17Fh
Bank3	1F0h-1FFh

Figura 1.6. Registros de propósito general de la memoria RAM⁵

También se muestra el mapa de memoria de datos.

¹⁵ <http://ciecfie.epn.edu.ec/PROYECTOS/pdfs/Explorador1.pdf>

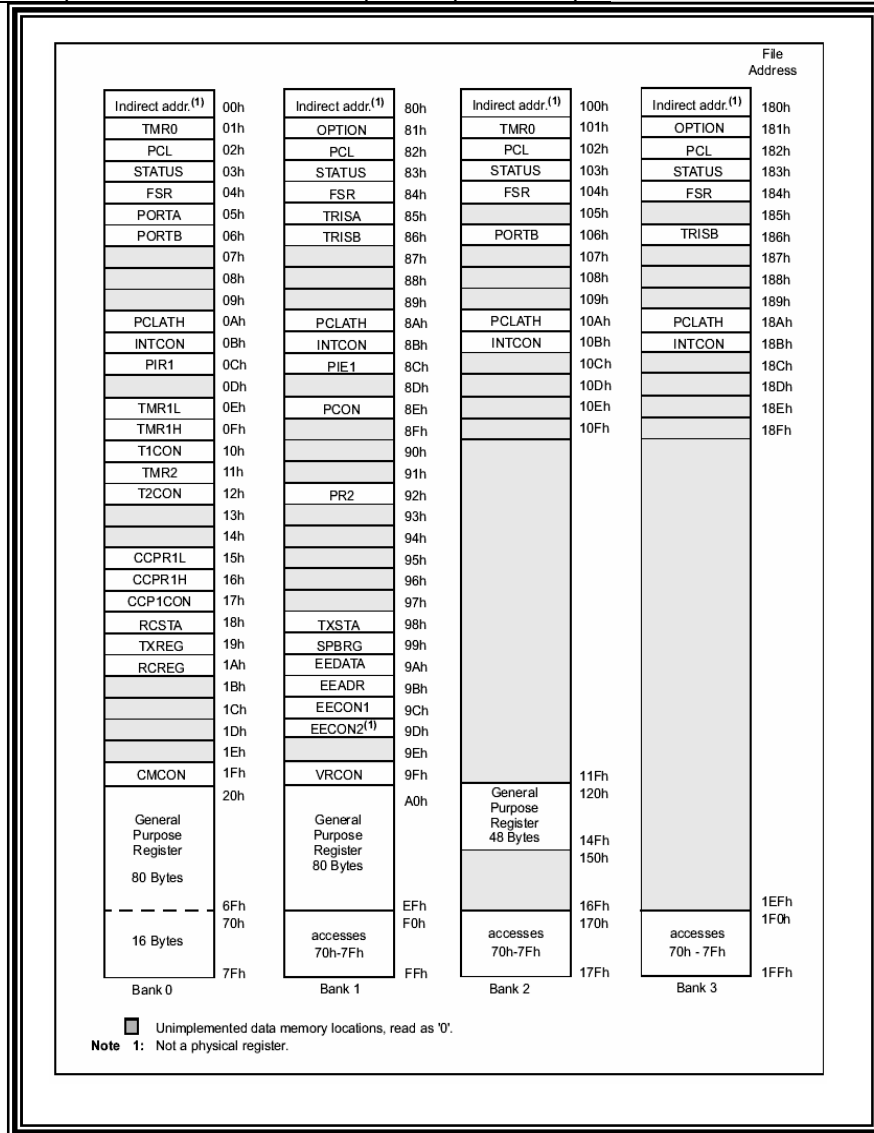


Figura 1.7. Mapa de memoria de datos⁵

1.1.4. LOS PUERTOS DEL PIC¹⁵

El PIC16F628A poseen dos puertos, los cuales pueden funcionar tanto de entrada como de salida: el puerto “A” de 8 terminales y el “B” de igual manera. Cada terminal puede estar predispuesto por el programa para operar como de entrada o de salida. Cada pin tiene un resistor de pull-up (resistor conectado a

fuente) interno que puede ser desconectado mediante el programa. Estos resistores se desconectan automáticamente si un pin se predispone como pin de salida debido a que las salidas ya tienen posibilidad de entregar corriente desde fuente con un transistor. Todos los resistores de pull-up se conectan o desconectan al mismo tiempo (no existe un comando que los conecte independientemente).

Como puerto de salida, un pin puede tomar 25mA del circuito o entregar 20mA al mismo, sin embargo, en el puerto "A" sólo se pueden consumir 80mA en total o entregar 50mA, esto significa que sólo algunos pines pueden trabajar al máximo porque si todos lo hicieran (y son 8) el consumo total sería de $25 \times 8 = 200\text{mA}$.

El puerto "B" tiene otras características máximas, ya que en total puede tomar 150mA o entregar 100mA, las salidas admiten suficiente carga como para alimentar directamente a un led (Figura 1.7.).

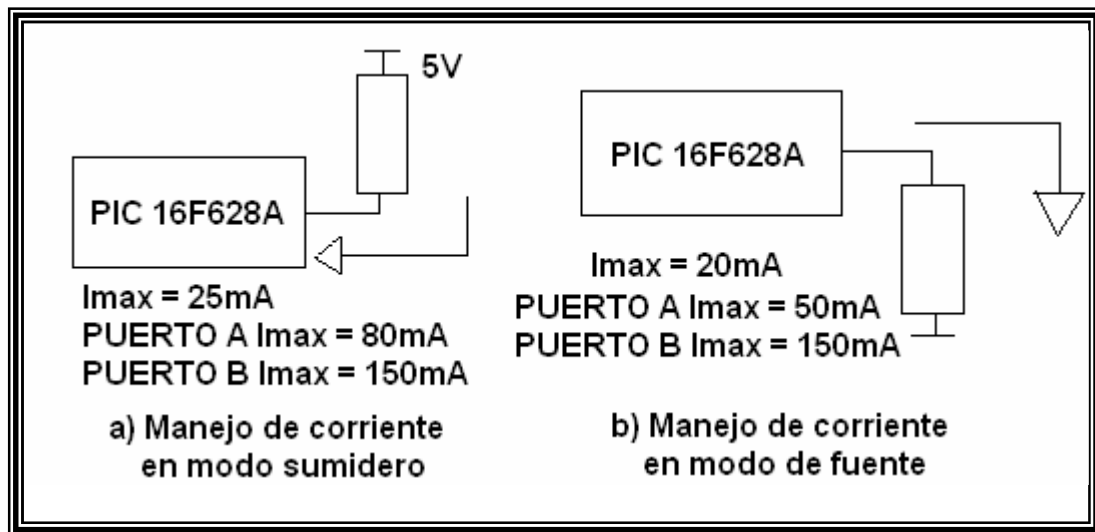


Figura 1.8. Manejo de corriente de los puertos del PIC ¹⁵

Los puertos no utilizados siempre se deben conectar a la fuente de 5V a través de un resistor de 10k, debido a que se trata de un dispositivo CMOS que, de otro modo, podría deteriorarse por captación electrostática.

El pin 3 perteneciente al puerto "A" puede ser configurada como de entrada-/salida o como de arranque de un temporizador/ contador. Cuando se programa

como entrada este pin funciona como un disparador de SCHMITT o Schmitt trigger ideal para reconocer señales distorsionadas o con crecimiento lento.

Este mismo pin también tiene una característica distinta cuando opera como salida. El es el único que trabaja a colector abierto, es decir, que no puede emplearse como fuente, en este caso siempre se utiliza un resistor externo.

El Clock: Los PIC poseen un oscilador configurable por programa de características muy amplias. Cuando no se requiere mucha precisión se puede trabajar con un oscilador interno, para circuitos que requieran una gran precisión se puede trabajar con un cristal de frecuencia baja, media o alta, como máximo el PIC16F628A puede trabajar con un cristal de 20Mhz. Internamente la frecuencia del cristal se divide por 4, por lo tanto, es muy común la utilización de un cristal de 4MHz para obtener un CLOCK interno de 1MHz que garantiza que cada instrucción dure exactamente 1mS. Para temporizadores de período largo se utilizan cristales de baja frecuencia.

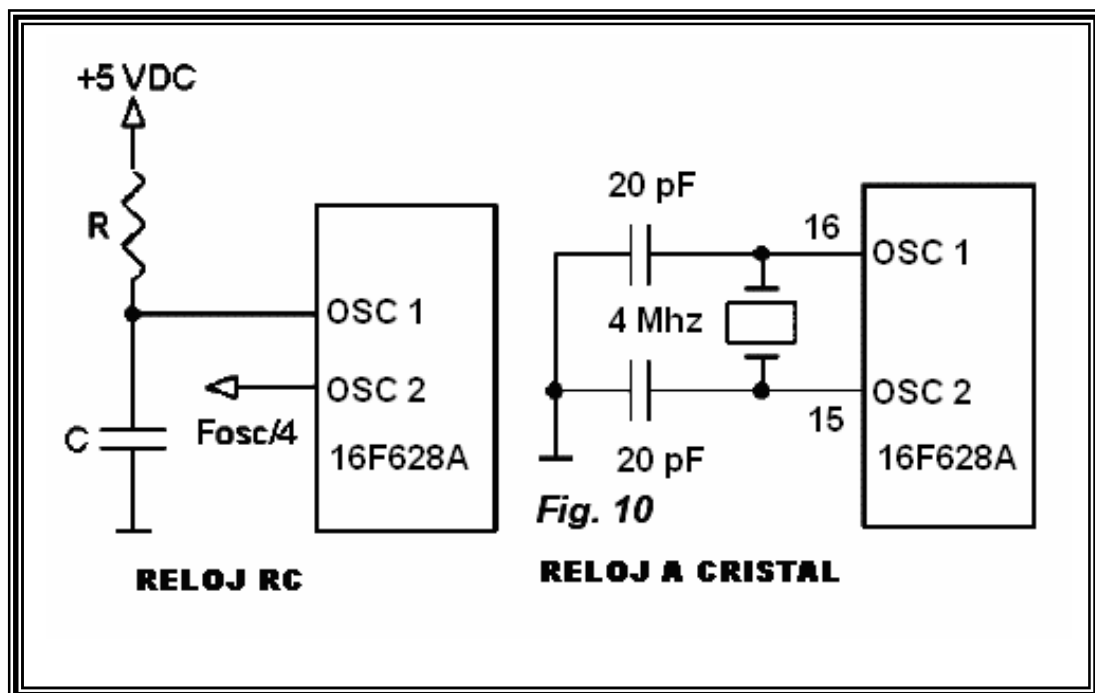


Figura 1.9. Tipos de Reloj externos ¹⁵

El Reset: El PIC "se resetea" cuando el pin 4 (MCLR negada) se pone a potencial bajo, el circuito de reset del PIC posee un temporizador interno que permite realizar un reset automático cuando se aplica tensión de 5V. En estos

casos el circuito externo de reset sólo implica el uso de un resistor de 10k entre la pin 4 y fuente.

En muchos circuitos es necesario realizar un reset manual y para ello existen dos posibilidades, una es utilizar sólo el temporizador interno (por programa) y la otra es agregar una constante de tiempo exterior.

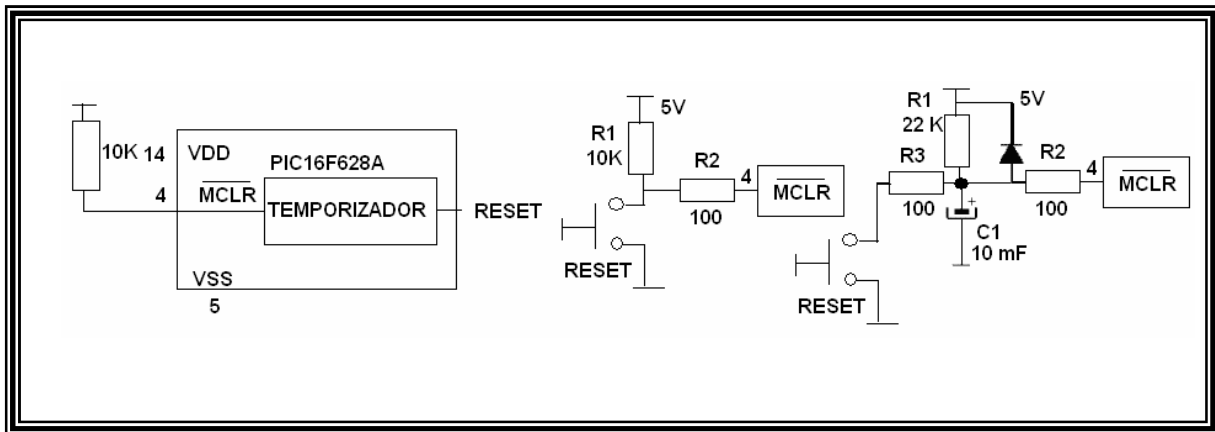


Figura 1.10. Circuitos de reset ¹⁵

1.1.5. SET DE INSTRUCCIONES DEL PIC 16F628A ¹⁴

A continuación se muestra todo el set de instrucciones o más conocido como el manual de referencia, para toda la familia de PICs y su correcta utilización, cabe destacar que los tiempos aquí descritos son tiempos virtuales utilizados para simular en el programa *PIC SIMULATOR IDE*.

1.1.5.1. Lista de todos los comandos Basic del compilador ¹⁴

SET DE INSTRUCCIONES	UTILIZACIÓN
<u>DIM</u>	Definir variables
<u>RESERVE</u>	Cambiar de variable
+, -, *, /	Operaciones aritméticas
<u>NOT</u> , <u>AND</u> , <u>OR</u> , <u>XOR</u> , <u>NAND</u> , <u>NOR</u> ,	Operaciones Lógicas
<u>XOR</u>	
<u>GOTO</u>	Ir hacia alguna etiqueta
<u>WAITMS</u> , <u>WAITUS</u>	Tiempos en milisegundos o microsegundos
<u>READ</u> , <u>WRITE</u>	Acceso a la memoria EEPROM
<u>FOR</u> , <u>TO</u> , <u>STEP</u> , <u>NEXT</u> , <u>WHILE</u> ,	Condicionantes
<u>WEND</u> , <u>IF</u> , <u>THEN</u> , <u>ELSE</u> , <u>ENDIF</u>	
<u>LOOKUP</u>	Seleccionar constantes
<u>SHIFTL</u> , <u>SHIFTR</u>	Desplazar niveles de bits
<u>ADCIN</u> , <u>ADC_CLOCK</u> ,	Para utilizar el convertidor de análogo a digital
<u>ADC_SAMPLEUS</u>	

<u>END</u>	Fin Del Programa
<u>GOSUB</u>	Ir y volver de alguna etiqueta
<u>RETURN</u>	Regreso de una subrutina
<u>ON INTERRUPT, RESUME</u>	Rutina de interrupción
<u>SAVE SYSTEM, ENABLE, DISABLE</u>	Control del bit GIE en el registro INTCON
<u>LCD BITS, LCD DREG, LCD DBIT,</u> <u>LCD RSREG, LCD RSBIT,</u> <u>LCD EREG, LCD EBIT,</u> <u>LCD RWREG, LCD RWBIT,</u> <u>LCD COMMANDUS, LCD DATAUS,</u> <u>LCD INITMS, LCDINIT, LCDOUT,</u> <u>LCDCMDOUT, LCDCLEAR,</u> <u>LCDHOME, LCDLINE2HOME,</u> <u>LCDLEFT, LCDRIGHT,</u> <u>LCDSHIFTLEFT, LCDSHIFTRIGHT,</u> <u>LCDLINE1CLEAR, LCDLINE2CLEAR,</u> <u>LCDLINE1POS, LCDLINE2POS,</u> <u>LCDDEFCHAR</u> <u>HSEROPEN,</u> <u>ALLOW MULTIPLE HSEROPEN,</u> <u>ALLOW ALL BAUDRATES,</u> <u>HSEROUT, HSERIN, HSERGET, LF,</u> <u>CRLF, SEROUT, SERIN,</u> <u>SEROUTINV, SERININV,</u> <u>SEROUT_DELAYUS, I2CWRITE,</u> <u>I2CREAD, I2CREAD_DELAYUS,</u> <u>I2CWRITE1, I2CREAD1,</u> <u>I2CPREPARE, I2CSTART, I2CSTOP,</u> <u>I2CSEND, I2CRECA,</u> <u>I2CRECEIVEACK, I2CRECN,</u> <u>I2CRECEIVENACK,ASM</u>	Configurar o inicializar el LCD
	Instrucciones para comunicación serial
	Comunicación Serial Sincrónica

Figura 1.11. Set de instrucciones ¹⁴

1.1.5.2. Utilización del set de instrucciones ¹⁴

Instrucción DIM: Las declaraciones pueden ponerse en cualquier parte del programa. Todas las variables son consideradas globales. El número total de variables está limitado por la memoria RAM disponible del microcontrolador, existen tres tipos de variables como el bit (1, 0), byte (enteros en el rango de 0 a 255), Word (2 byte, enteros en el rango de 0 a 65535). Ejemplo:

DIM A AS BIT

DIM B AS BYTE

DIM X AS WORD

¹⁴ <http://www.oshonsoft.com/downloads.html>

Instrucción RESERVE: Permite el uso avanzado de la memoria reservando algunas localidades de la RAM para que sean usadas por el código assembler de las rutinas o por MPLAB en la depuración. Por ejemplo: RESERVE 0x70.

Todos los registros de funciones especiales (SFRs) están disponibles como variables tipo Byte en los programas Basic. Los bits individuales de una variable Byte pueden ser direccionados por las extensiones .0, .1, .2, .3, .4, .5, .6 y .7 o usando los nombres oficiales de los bits:

DIM A AS BIT

DIM B AS BYTE

A = B.7

B.6 = 1

TRISA.1 = 0

TRISB = 0

PORTA.1 = 1

PORTB = 255

STATUS.RP0 = 1

INTCON.INTF = 0

Las formas cortas Standard para acceder a los pórtilos o pines del chip están disponibles (RA, RB, RC, RD, RE pueden usarse como variables Byte; RA0, RA1, RA2,..., RE6, RE7 están disponibles como variables Bit).

Operaciones aritméticas: Cinco operaciones aritméticas (+, -, *, /, MOD) están disponibles para datos tipo Byte o Word. El compilador está habilitado para compilar todas las expresiones aritméticas complejas. Por ejemplo:

DIM A AS WORD

DIM B AS WORD

A = 123

B = A * 234

X = (12345 - B * X) / (A + B)

La raíz cuadrada de un número se calcula usando la función SQR:

DIM A AS WORD

A = 3600

A = SQR(A)

Operaciones Lógicas: Para variables de datos tipo bit están disponibles siete operaciones lógicas. Es posible realizar solamente una operación lógica en cada instrucción. Las operaciones lógicas también están disponibles para variables Byte y Word. Por Ejemplo:

DIM A AS BIT

DIM B AS BIT

DIM X AS BIT

X = NOT A

X = A XOR B

X = A NAND B

X = A NOR B

X = A NXOR B

DIM A AS WORD

DIM B AS WORD

A = A OR B

PORTB = PORTC AND %11110000

Instrucción GOTO: La instrucción GOTO usa el nombre de la etiqueta de línea como argumento. La etiqueta de línea debe estar seguida por dos puntos (:).
Ejemplo:

DIM A AS WORD

A = 0

```
loop: A = A + 1  
GOTO loop
```

Instrucciones de tiempo: Las instrucciones WAITMS y WAITUS pueden usarse para forzar al programa que espere por el número especificado de milisegundos o microsegundos. Se pueden también usar como argumentos a variables de tipos de dato Byte o Word. Estas rutinas usan parámetros de frecuencia de reloj que pueden cambiarse desde el menú Options. La rutina WAITUS tiene mínimos retardos y pasos que también dependen del parámetro de la frecuencia de reloj.

```
DIM A AS WORD  
A = 100  
WAITMS A  
WAITUS 50
```

Cuando escribimos programas para PICs usamos intervalos de retraso de alrededor de 1 segundo o 1000 milisegundos. Muchos ejemplos usan intervalos en “tiempo real”; aunque actualmente haya disponibles poderosas PCs. Para la simulación de la instrucción 'WaitMs 1000' a 4MHz en el simulador deben realizarse 1000000 de instrucciones y esto toma un tiempo considerable aunque se haya seleccionado el rango de 'extremely fast' en la simulación. Así, solo para el propósito de simulación se debería recompilar sus programas con intervalos de retardo reajustados (provisionales), que no deberían exceder 1 a 10 ms. Pero, hay que asegurarse de recompilar los programas con los retardos de tiempo originales antes de grabar el PIC.

Memoria EEPROM: El acceso a la memoria de datos EEPROM puede hacerse usando las instrucciones READ y WRITE. El primer argumento es la dirección de un byte en la memoria EEPROM y puede ser una constante o una variable Byte. El segundo argumento es el dato que es leído o escrito (para la instrucción READ este argumento debe ser una variable Byte). Se debe guardar interrupciones deshabilitadas durante la ejecución de la instrucción WRITE.

```
DIM A AS BYTE
DIM B AS BYTE
A = 10
READ A, B
WRITE 11, B
```

Condicionantes: Tres instrucciones del BASIC standard están disponibles: FOR-TO-STEP-NEXT, WHILE-WEND and IF-THEN-ELSE-ENDIF. Ejemplos:

Ejemplo de WHILE:

```
DIM A AS BYTE
TRISB = 0
A = 255
WHILE A > 0
PORTB = A
A = A - 1
WAITMS 100
WEND
PORTB = A
```

```
TRISB = 0
loop:
IF PORTA.0 THEN
    PORTB.0 = 1
ELSE
    PORTB.0 = 0
ENDIF
GOTO loop
```

Ejemplo de FOR-NEXT:

```
DIM A AS WORD
TRISB = 0
FOR A = 0 TO 10000 STEP 10
```

```

    PORTB = A.LB
NEXT A

DIM A AS BYTE
DIM B AS BYTE
DIM X AS BYTE
B = 255
X = 2
TRISB = 0
FOR A = B TO 0 STEP -X
PORTB = A
NEXT A

```

Instrucción LOOKUP: Esta puede usarse para seleccionar una constante de una lista de constantes tipo Byte, la ubicación de la constante seleccionada viene dada por el valor de otra variable tipo Byte. La primera constante en la lista tiene un índice de valor 0. La constante seleccionada puede cargarse en una variable de datos tipo Byte. Si el valor de la variable índice es mayor que el número de constantes en la lista, el resultado obtenido es el mismo que se obtendría cuando el valor de la variable índice es igual al número de constantes en la lista. Ejemplo que puede verse en un display de 7 segmentos:

```

DIM DIGIT AS BYTE
DIM MASK AS BYTE
loop:
TRISB = %00000000
FOR DIGIT = 0 TO 9
    MASK = LOOKUP(0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F,
0x6F), DIGIT
    PORTB = MASK
    WAITMS 1000
NEXT DIGIT
GOTO loop

```

Desplazar bits: Las funciones SHIFTLEFT y SHIFTRIGHT pueden usarse para desplazar niveles de bits de una variable hacia la izquierda o hacia la derecha. El primer argumento es la variable de entrada y el segundo argumento es el número de deslizamientos a desarrollarse. Ejemplos:

```
TRISB = 0x00
```

```
PORTB = %00000011
```

```
goleft:
```

```
WAITMS 250
```

```
PORTB = SHIFTLEFT (PORTB, 1)
```

```
IF PORTB = %11000000 THEN GOTO goright
```

```
GOTO goleft
```

```
goright:
```

```
WAITMS 250
```

```
PORTB = SHIFTRIGHT (PORTB, 1)
```

```
IF PORTB = %00000011 THEN GOTO goleft
```

```
GOTO goright
```

```
TRISB = 0x00
```

```
PORTB = %00000001
```

```
goleft:
```

```
WAITMS 250
```

```
PORTB = SHIFTLEFT (PORTB, 1)
```

```
IF PORTB.7 THEN goright
```

```
GOTO goleft
```

```
goright:
```

```
WAITMS 250
```

```
PORTB = SHIFTRIGHT (PORTB, 1)
```

```
IF PORTB.0 THEN goleft
```

```
GOTO goright
```


Convertidor A/D: La instrucción ADCIN está disponible como un soporte para el convertidor A/D interno. El primer argumento es el número del canal ADC y el segundo argumento es una variable que será usada para guardar el resultado de la conversión A/D. La instrucción ADCIN usa dos parámetros ADC_CLOCK y ADC_SAMPLEUS que tienen por defecto valores 3 y 20. Estos valores por defecto pueden cambiarse usando la directiva DEFINE. El parámetro ADC_CLOCK determina la fuente de reloj escogida para el ADC (el rango permitido es 0-3 o 0-7 dependiendo del dispositivo usado). El parámetro ADC_SAMPLEUS setea el tiempo de adquisición deseado del ADC en milisegundos (0-255). La instrucción ADCIN propone el pin correspondiente estar configurado como una entrada análoga (con los registros TRIS, ADCON1 y en algunos dispositivos con el registro ANSEL). Ejemplo:

```
DIM V (5) AS BYTE
DIM VM AS WORD
DIM I AS BYTE
DEFINE ADC_CLOCK = 3
DEFINE ADC_SAMPLEUS = 50
TRISA = 0xFF
TRISB = 0

ADCON1 = 0
FOR I = 0 TO 4
  ADCIN 0, V (I)
NEXT I
VM = 0
FOR I = 0 TO 4
  VM = VM + V (I)
NEXT I
VM = VM / 5
PORTB = VM.LB
```

Instrucción GOSUB: Los programas estructurados pueden escribirse usando llamadas a subrutinas con la instrucción GOSUB que usan nombres de etiquetas de línea como argumento. El regreso desde una subrutina se realiza

con la instrucción Return. Cuando se usa subrutinas, el programa principal (llamado también rutina principal) necesita finalizar con la instrucción END. La instrucción END se compila como un lazo infinito. Ejemplo:

```
SYMBOL ad_action = ADCON0.GO_DONE
SYMBOL display = PORTB
TRISB = %00000000
TRISA = %1111111
ADCON0 = 0xC0
ADCON1 = 0
HIGH ADCON0.ADON
main:
GOSUB getadresult
display = ADRESH
GOTO main
END
getadresult:
HIGH ad_action
WHILE ad_action
WEND
RETURN
```

Interrupción: La rutina de interrupción debe ser ubicada como cualquier otra subrutina después de la instrucción END. Debe empezar con ON INTERRUPT y terminar con la instrucción RESUME. Si operaciones aritméticas, arreglos o alguna otra instrucción compleja son usadas en la rutina de interrupción, entonces la instrucción SAVE SYSTEM debe ponerse a continuación de la instrucción ON INTERRUPT para grabar el contenido de los registros usados por el sistema. Las instrucciones ENABLE y DISABLE pueden usarse en el programa principal para el control del bit GIE en el registro INTCON. La instrucción RESUME setea el bit GIE y habilita nuevas interrupciones. Por ejemplo:

```
DIM A AS BYTE
A = 255
```

```

TRISA = 0
PORTA = A
INTCON.INTE = 1
ENABLE
END
ON INTERRUPT
    A = A - 1
    PORTA = A
    INTCON.INTF = 0
RESUME

DIM T AS WORD
T = 0
TRISA = 0xFF
ADCON1 = 0
TRISB = 0
OPTION_REG.T0CS = 0
INTCON.T0IE = 1
ENABLE
loop:
    ADCIN 0, PORTB
GOTO loop
END
ON INTERRUPT
    SAVE SYSTEM
    T = T + 1
    INTCON.T0IF = 0
RESUME

```

Manejo del LCD: El compilador Basic también sirve para manejar módulos LCD. Antes de usar las instrucciones relacionadas con el LCD, se debe setear la interfaz del LCD usando las directivas DEFINE, de la siguiente manera:

LCD_BITS - define el número de líneas del bus de datos (se permite los valores 4 y 8; por defecto es 4).

LCD_DREG - define el p rtico donde se conecta el bus de datos (por defecto es PORTB)

LCD_DBIT – define la posici n de las l neas de datos para la interfaz de 4-bits (0 o 4; por defecto es 4), se ignora para la interfaz de 8-bits

LCD_RSREG - define el p rtico donde se conecta la l nea RS (por defecto es PORTB).

LCD_RSBIT - define el pin a donde se conecta la l nea RS (por defecto es 3).

LCD_EREG - define el p rtico a donde se conecta la l nea E line (por defecto es PORTB)

LCD_EBIT – define el pin a donde se conecta la l nea E (por defecto es 2)

LCD_RWREG – define el p rtico a donde se conecta la l nea R/W (setear a 0 si no se usa; 0 es por defecto)

LCD_RWBIT – define el pin a donde se conecta la l nea R/W (setear a 0 si no se usa; 0 es por defecto).

LCD_COMMANDUS - define el retardo despu s de la instrucci n.

LCDCMDOUT (por defecto el valor es 5000).

LCD_DATAUS - define el retardo despu s de la instrucci n LCDOUT (por defecto el valor es 50).

LCD_INITMS - define el retardo para la instrucci n LCDINIT (el valor por defecto es 100).

Los  ltimos tres par metros deben setearse a valores bajos cuando se use el m dulo LCD del simulador.

La instrucci n LCDINIT debe ponerse en el programa antes de cualquier instrucci n LCDOUT (usada para enviar datos) y LCDCMDOUT (usada para enviar comandos). Su argumento se usa para definir el tipo de cursor: 0 = no

cursor (por defecto), 1 = blink, 2 = subrayado, 3 = blink + subrayado. Las instrucciones LCDOUT y LCDCMDOUT pueden tener múltiples argumentos separados por ','. Strings, constantes y variables pueden usarse como argumentos de la instrucción LCDOUT. Si se usa el signo '#' antes del nombre de una variable entonces su representación decimal se envía al módulo LCD.

Las Constantes y variables pueden usarse como argumentos, la instrucción LCDCMDOUT y las siguientes palabras reservadas también están disponibles: LcdClear, LcdHome, LcdLine2Home, LcdLeft, LcdRight, LcdShiftLeft, LcdShiftRight, LcdLine1Clear, LcdLine2Clear, LcdLine1Pos () y LcdLine2Pos (). El argumento de LcdLine1Pos () y LcdLine2Pos () puede ser un número en el rango (1-40) o una variable de datos tipo Byte. El valor contenido en esta variable debe estar en el mismo rango. Ejemplos

```
DEFINE LCD_BITS = 8           ; 8 líneas de datos
DEFINE LCD_DREG = PORTB      ; salida de datos por el PORTB
DEFINE LCD_DBIT = 0          ; posición 0 de línea de datos
DEFINE LCD_RSREG = PORTD     ; RS se conecta al PORTD
DEFINE LCD_RSBIT = 1         ; RS se conecta al pin PORTD.1
DEFINE LCD_EREG = PORTD     ; enable se conecta al PORTD
DEFINE LCD_EBIT = 3          ; E se conecta al PORTD.3
DEFINE LCD_RWREG = PORTD     ; lectura/escritura se conecta al PORTD
DEFINE LCD_RWBIT = 2         ; R/W se conecta al PORTD.2
DEFINE LCD_COMMANDUS = 10000 ; retardo de 10000 µs
DEFINE LCD_DATAUS = 100      ; retardo después de LCDOUT 100 µs
DEFINE LCD_INITMS = 1000     ; retardo después de LCDINIT 1000 µs
LCDINIT                       ; para envió de datos
loop:                          ; etiqueta loop
    LCDOUT "Hello world!"     ; sacar en LCD hola mundo
    WAITMS 1000               ; retardo de 1000 ms
    LCDCMDOUT LcdClear        ; para envió de comandos
    WAITMS 1000               ; retardo de 1000 ms
GOTO loop                     ; retornar a lazo loop
```

Las instrucciones descritas para el LCD tomarán el mando sobre los registros TRIS conectados con los pines usados para la interfaz con el LCD, pero si se usa pines de PORTA o PORTE en dispositivos con el módulo convertidor A/D entonces se debe tomar el control sobre el registro ADCON1 para setear los pines usados como I/O digitales.

Se puede instalar hasta ocho caracteres definidos a usarse en el LCD. Esto se lo puede hacer fácilmente con la instrucción LCDDEFCHAR.

Comunicación Serial: El soporte tanto para hardware como para software en la comunicación serial está disponible con las instrucciones HSEROPEN, HSEROUT, HSERIN y HSERGET pueden usarse con los PICs que tengan hardware UART interno. La instrucción HSEROPEN setea el hardware UART. Su único argumento es el rango del baud y los valores a escoger pueden ser: 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 31250, 38400 y 57600. Si se omite el argumento el UART será seteado para un rango de 9600 baud. Si el parámetro ALLOW_MULTIPLE_HSEROPEN se setea a 1 usando la directiva DEFINE, esto hará posible usar la instrucción HSEROPEN más de una vez en el programa, por ejemplo para cambiar la proporción baud seleccionada. Si el parámetro ALLOW_ALL_BAUDRATES es seteado a 1 usando la directiva DEFINE todos las proporciones de baud en el rango 100-57600 se permitirán. La instrucción HSEROUT se usa para transmisión serial. La instrucción HSEROUT puede tener múltiples argumentos separados por ','. Se puede usar strings, LF keyword para alimentación de caracteres por línea o CRLF keyword para Retorno de carro – Secuencia de alimentación de línea, constantes y variables. Si se usa el signo '#' antes del nombre de una variable entonces su representación decimal es enviada al pósito serial. La instrucción HSERIN puede usarse para cargar una lista de variables Byte y Word con los valores recibidos en el pósito serial. Esta instrucción esperará hasta que el número de bytes requeridos se reciban en el pósito serial.

La instrucción HSERGET tiene un argumento que debe ser una variable Byte. Si hay un carácter esperando en el buffer de recepción este será cargado en la variable, caso contrario el valor 0 será cargado. Ejemplos:

```

DIM I AS BYTE           ; definición de variable I
HSEROPEN 38400          ; datos seriales a 38400 baudios
WAITMS 1000            ; retardo de 1000 ms
FOR I = 20 TO 0 STEP -1 ; lazo for de 20 en pasos de -1
  HSEROUT "Number: ", #I, CrLf ; transmisión serial
  WAITMS 500           ; retardo de 500 ms
NEXT I                  ; continúe en el lazo de variable I

```

Para todos los dispositivos PIC que los soporten se puede usar software de rutinas de comunicación serial con las instrucciones SEROUT y SERIN. El primer argumento de ambas instrucciones deben ser uno de los pines del microcontrolador, y el segundo argumento es la proporción baud: 300, 600, 1200, 2400, 4800 o 9600. Para la instrucción SEROUT entonces sigue la lista de argumentos a ser enviados al p rtico serial. Puede usar strings, LF keyword para caracteres de alimentaci n de l nea (Line Feed character) o CRLF keyword para retorno de carro – secuencia de alimentaci n de l nea (Carriage Return - Line Feed sequence), constantes y variables. Si se usa el signo '#' antes del nombre de una variable entonces su representaci n decimal es enviada al p rtico serial. La instrucci n SEROUT usa el par metro SEROUT_DELAYUS que puede ser seteado por la directiva DEFINE y tiene por defecto el valor de 1000 microsegundos.

Este define el intervalo de retardo antes de que un car cter que actualmente es enviado al p rtico y este sea usado para aumentar la fiabilidad del software de la rutina SEROUT. Para la instrucci n SERIN entonces sigue la lista de variables Byte y Word a ser cargadas con los valores recibidos en el p rtico serial.

Esta instrucci n esperar  hasta que el n mero de bytes requerido sean recibidos en el p rtico serial. Para la interfaz serial con niveles de l gica invertida est n disponibles las instrucciones SERININV y SEROUTINV.
Ejemplos:

```

DEFINE SEROUT_DELAYUS = 5000
SEROUT PORTC.6, 1200, "Hello world!", CrLf

```

DIM I AS BYTE

loop:

SERIN PORTC.7, 9600, I

SEROUT PORTC.6, 9600, "Number: ", #I, CrLf

GOTO loop

1.1.6. GRABADORES DE PICS ¹⁵

Para que un PIC funcione es preciso “cargarle” un programa en la “memoria de programa”. El programa debe estar en un lenguaje comprensible por el chip o sea un lenguaje de máquina en código binario aunque en realidad al PIC se debe introducirle un programa en código hexadecimal e internamente lo convierte a binario para realizar su interpretación.

El programa que se debe cargarle al PIC en código hexadecimal debe tener la terminación “hex”. Sin embargo, normalmente se debe escribir, el programa en un lenguaje entendible por el usuario, normalmente conocido como “assembler”. Los PIC basan su programación en un set reducido de instrucciones que permiten editar el programa generando un archivo con extensión “asm”. Existen programas que se encargan de “traducir” el programa escrito en lenguaje assembler (asm), a programa escrito en lenguaje de máquina (hex); a estos programas se los denomina traductores o ensambladores. Por lo tanto, para escribir un programa y cargarlo en un PIC se necesita; un utilitario o programa que nos permita escribir el juego de sentencias a cargar en la memoria de programa del PIC. Este utilitario es un “editor” y como ejemplo tenemos el MPLAB.

Un utilitario o programa que permita “traducir” el programa escrito en assembler a lenguaje de máquina para que se pueda cargarlo en el PIC. A ese programa se lo denomina “ensamblador” y como ejemplo se tiene el MPASM.

Un utilitario para “cargar” el archivo que tiene el programa en lenguaje de máquina (archivo con extensión hex) en la memoria del PIC. A estos

programas se los denomina “cargadores” y como ejemplo se tiene al NOPPP, al PIC PRO.

Pero también existen otros utilitarios que permiten “simular” y “emular” un programa para ver cómo se comporta antes de cargarlo en el PIC.

El simulador permite detectar errores en el programa (el MPLAB permite simular) para poder corregirlos. El emulador permite que “una computadora se comporte como PIC” siendo los pines del puerto, los correspondientes a los del PIC. Esto quiere decir que si se desea realizar circuitos con PICs, para poder disponerlo adecuadamente se necesita:

Programa MPLAB: Para editar (escribir) un programa en lenguaje assembler y así generar el archivo con *extensión asm*. El MPLAB también permite “simular” el funcionamiento del programa para detectar errores y corregirlos.

Programa MPASM: Para “ensamblar” o convertir el archivo asm en archivo hex con el objeto de cargarlo en el PIC.

Programa NOPPP: O cualquier otro cargador que permita ingresar el archivo con extensión hex en la memoria de programa del PIC. También se puede emplear cualquier otro cargador, hasta incluso se podrían cargar los datos manualmente. Todos los programas grabadores necesitan de un circuito “electrónico” o hardware para poder realizar la función de grabación.

Los Cargadores De Pics: Un cargador de PIC debe poseer por un lado el circuito que permita grabar el programa desde una computadora (hardware) y el programa que permita manejar a dicho circuito para realizar la operación de grabación (software).

Un dispositivo para una PC (hardware), siempre necesita un programa que lo controle (software), es posible obtener un software gratuito llamado NOPPP que se puede bajar del Internet, el MPLAB, el MPASM y otros programas utilitarios, en la Web www.webelectronica.com.ar.

Cuando se dice “programar” se refiere a diseñar un programa para un PIC “programador”; y al decir “grabar” se refiere a llenar con datos la memoria de un PIC y el dispositivo que realiza esta función es el “grabador de PICs”. Los circuitos grabadores son también conocidos como cargadores de PICs y son los siguientes:

- * Cargador NOPPP
- * Cargador PROPIC
- * Cargador Universal de PICs y EEPROM
- * Cargador PIC800
- * Cargador PP84
- * WINPIC 800 3.55G.

Estos cargadores sirven para ser utilizados en diferentes tipos de PICs a saber:

NOPPP es un programa de uso libre que sirve para cargar PICs del tipo 16C84, 16F83 y 16F84, pero también se muestran otros cargadores con sus respectivos programas, que sirven para otros componentes.

PICPRO es un programador avanzado en relación con el NOPPP y sirve para ser montado cuando no se quieren correr riesgos con la PC que se está utilizando para cargar PICs y cuando se desea tener un trabajo libre de errores (el NOPPP es un programa para aprendizaje).

El Programador Universal: sirve para programar: 12C508, 12C509, 12C67X, 16C55X, 16C61, 16C62X, 16C71, 16C71X, 16C8X, 16F8X, 16F873/4/6/7, 16C73/74/76/77. También programa memorias EEPROM de la serie 24LCXX.

PIC800: es un programador para 16C84, 16F84, 16F873/74/76/77.

PP84: es un programador sencillo del tipo NOPPP de buenas prestaciones.

Un circuito con PIC siempre se compone de dos secciones. El circuito del dispositivo deseado y los componentes necesarios para que el PIC funcione, a

saber clock y reset (la primer sección cambia con cada proyecto, en tanto que la segunda puede ser común a muchos proyectos). Ambos circuitos periféricos tienen diferentes opciones: el clock puede ser el interno, a cristal cuando se requiera precisión, o a RC cuando sólo se requiere que la secuencia de encendido tenga un tiempo aproximado. En cuanto al reset, éste puede ser externo con un pulsador o interno y automático, cada vez que se conecta la alimentación de 5V, Ejemplo:

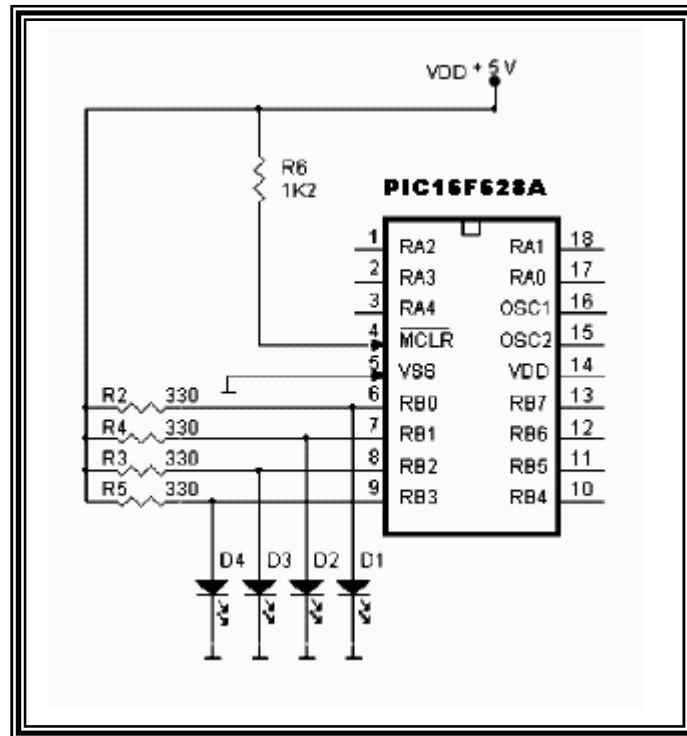


Figura 1.12. Conexiones del PIC¹⁵

El clock se utiliza el interno del PIC, en tanto que el reset se realiza en forma automática, cada vez que se enciende el equipo, por intermedio de R6, también se utiliza 4 LEDs sobre los pines RA0 a RA3 puerto A y sus correspondientes resistores de pull up (resistores a fuente).

Grabador para puerto paralelo: El "Programador de PICs" o también conocido como "Cargador de PICs", se tiene varios, el primero que funciona con el puerto paralelo, puerto serial o el USB, luego se necesita el driver del dispositivo y un programa de aplicación que utilice dicho dispositivo. Estos cargadores de PICs se diferencian en la velocidad a la cual cargan el PIC; este funciona con un software gratuito llamado NOPPP, el cual puede ser adquirido a través del Internet.

El PIC, como una memoria, tiene un terminal que predispone el dispositivo para leer o para escribir. Si el terminal 4 del PIC está a un potencial comprendido entre 13 y 14 V, el PIC está preparado para escribir los datos que provienen del computador. Si el terminal 4 está por debajo de 6V, el dispositivo está previsto para ser leído. Los datos a leer o escribir se ponen/obtienen de el terminal 13 del PIC con el pin 12 del mismo que opera como reloj.

Un PIC se lee/escribe accediendo a las diferentes posiciones de memoria por el mismo terminal por el que se obtienen/ ingresan los datos (terminal 13). La señal primero elige la posición de memoria a ser leída/escrita, y luego que esa posición está accesible se escriben o leen los datos. El clock que se coloca en la terminal 12 sirve para indicar en qué momento se debe transferir la información.

Los datos pueden estar sobre el pin 13 todo el tiempo, ya que no serán ni leídos ni escritos por la PC hasta que se produzca un cambio de estado (de alto para bajo) en el terminal 12.

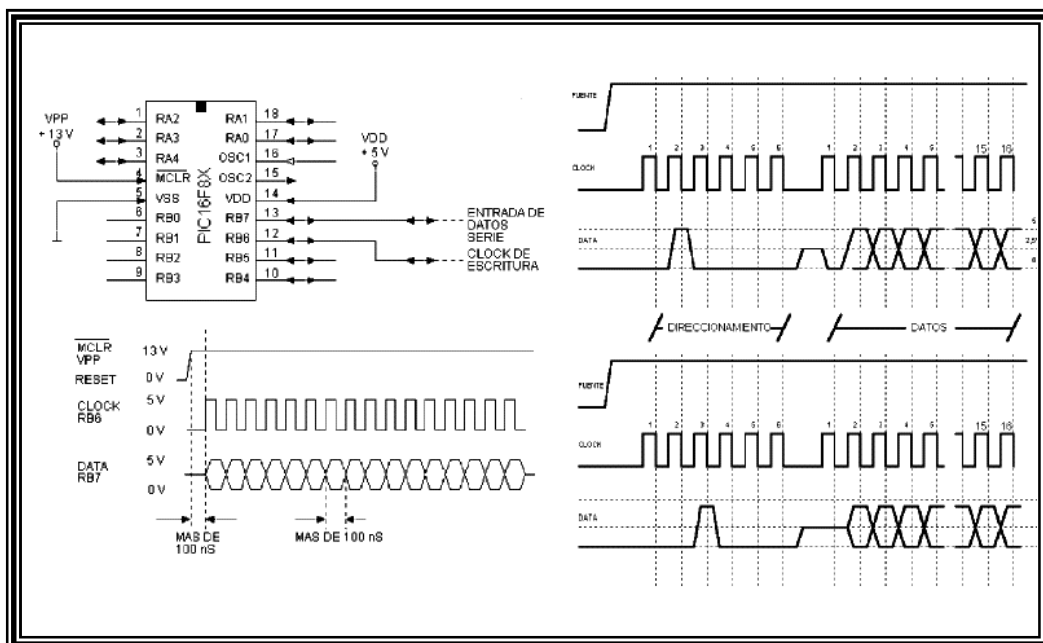
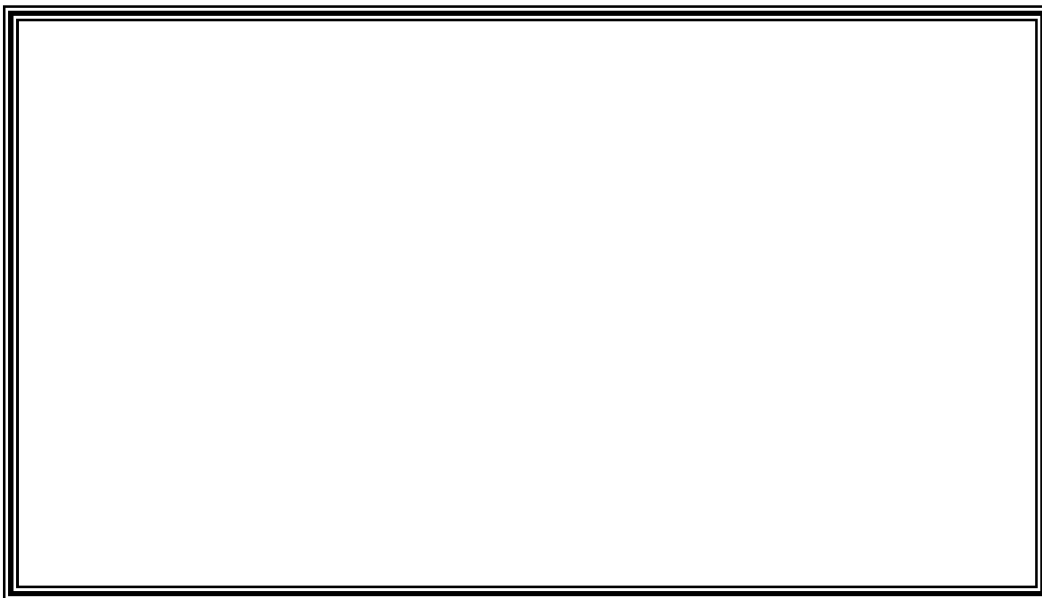


Figura 1.13. Señales de programación y lectura del PIC ¹⁵

Pasos para la grabación del PIC:

- a. El primer paso es colocar el PIC en el zócalo del Programador con señales y fuentes a potencial de masa.
- b. Levantar la tensión de fuente VDD a un potencial de $5V \pm 0,2V$ por el pin 14 (VDD).
- c. Levantar la tensión de fuente VPP a un potencial de $13V \pm 0,3V$ por el pin 4 (MCLR NEGADO).
- d. Esperar en esas condiciones un tiempo superior a 1mS.
- e. Posicionar el primer dato en el pin 13 (RB7) con un potencial alto (mayor a 4 V) o bajo (menor a 1V).
- f. Cuando el pin 12 (RB6) pase a un estado bajo, inferior a 1V, el dato se carga en la memoria.
- g. Continuar cargando los datos con el mismo criterio a un ritmo tal que el dato este presente por lo menos durante 100nS.
- h. Cuando todos los datos fueron cargados se debe esperar 1 segundo.
- i. Desconectar la fuente de 13V.
- j. Desconectar la fuente de 5V.
- k. Retirar el micro grabado.



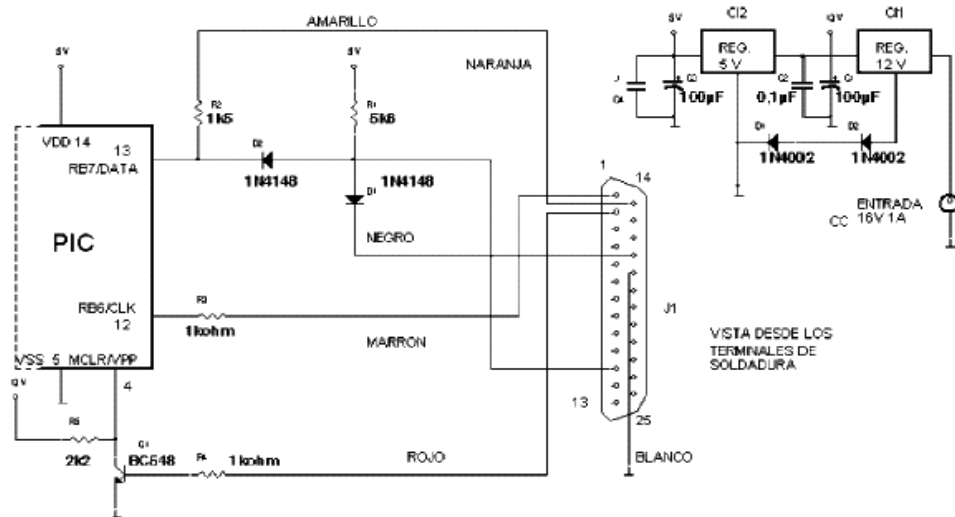


Figura 1.14. Circuito Esquemático del Programador para puerto paralelo ¹⁵

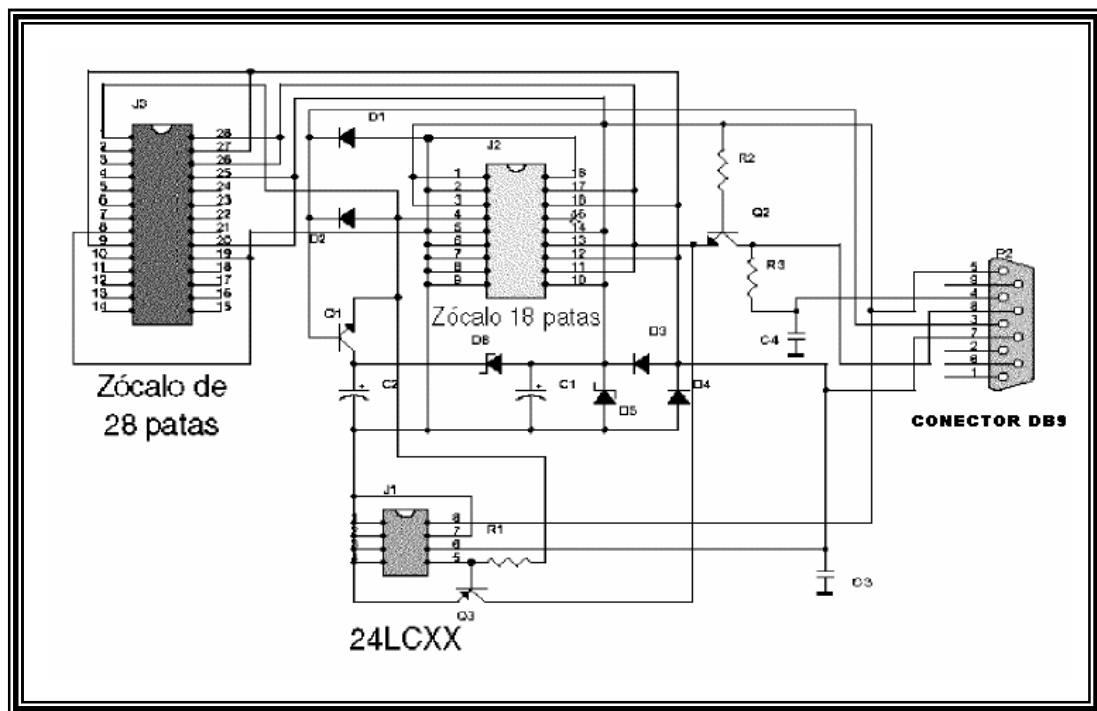


Figura 1.15. Circuito Esquemático del Programador para puerto serial ¹⁵

1.2. TIPOS DE DISPLAY 9X12CM ¹⁶

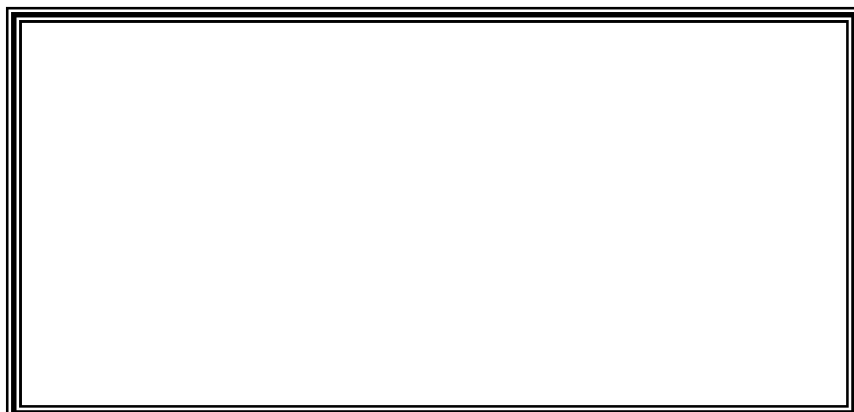




Figura 1.16. Display de 7 segmentos ¹⁶

El tipo más conocido de indicador alfanumérico es el display de siete segmentos, el cual comprende siete segmentos fotoeléctricos accesibles independientemente (tales como LEDs, elementos de cristal líquido, de descarga de gas, fluorescentes, etc.). Los segmentos son denominados convencionalmente de “a” a “g”, siendo posible hacer que muestren cualquier número del 0 al 9 o un carácter alfabético de la A a la F (mezclando letras mayúsculas y minúsculas) activando estos segmentos en distintas combinaciones, tal como se muestra en la (figura 1.16.).

Segmentos (✓ = ON)							Display	Segmentos (✓ = ON)							Display
a	b	c	d	e	f	g		a	b	c	d	e	f	g	
✓	✓	✓	✓	✓	✓		0	✓	✓	✓	✓	✓	✓	✓	8
	✓	✓					1	✓	✓	✓			✓	✓	9
✓	✓		✓	✓		✓	2	✓	✓	✓		✓	✓	✓	A
✓	✓	✓	✓			✓	3			✓	✓	✓	✓	✓	b
	✓	✓			✓	✓	4	✓			✓	✓	✓		c
✓		✓	✓		✓	✓	5		✓	✓	✓	✓		✓	d
✓		✓	✓	✓	✓	✓	6	✓			✓	✓	✓	✓	E
✓	✓	✓					7	✓				✓	✓	✓	F

Figura 1.17. Segmentos del display ¹⁶

El display de siete segmentos debe disponer como mínimo de ocho terminales de conexión exterior; siete de ellos proporcionan acceso a los segmentos fotoeléctricos individuales, y el octavo proporciona una conexión común a todos los segmentos. Si el display es de tipo LED, los ánodos de los LEDs deben

estar conectados a un terminal común, o en la que todos los cátodos de los LEDs están conectados a un terminal común.

En el primer caso, el dispositivo es conocido como display de siete segmentos de ánodo común, y en el segundo caso como display de siete segmentos de cátodo común. Los display de siete segmentos son utilizados para proporcionar una indicación visual de los estados de salida de circuitos integrados digitales tales como contadores de décadas, de bloqueo. Estas salidas normalmente están en forma de BCD (decimal codificado en binario) de cuatro bits y no son adecuadas para activar directamente los displays de siete segmentos, por consiguiente, deben colocarse decodificadores/excitadores especiales de conversión de BCD a siete segmentos, para convertir la señal BCD a una forma adecuada que active estos displays, los cuales están intercalados entre las señales BCD y el display.

¹⁶ <http://www.cypsela.es/especiales/pdf195/display.pdf>

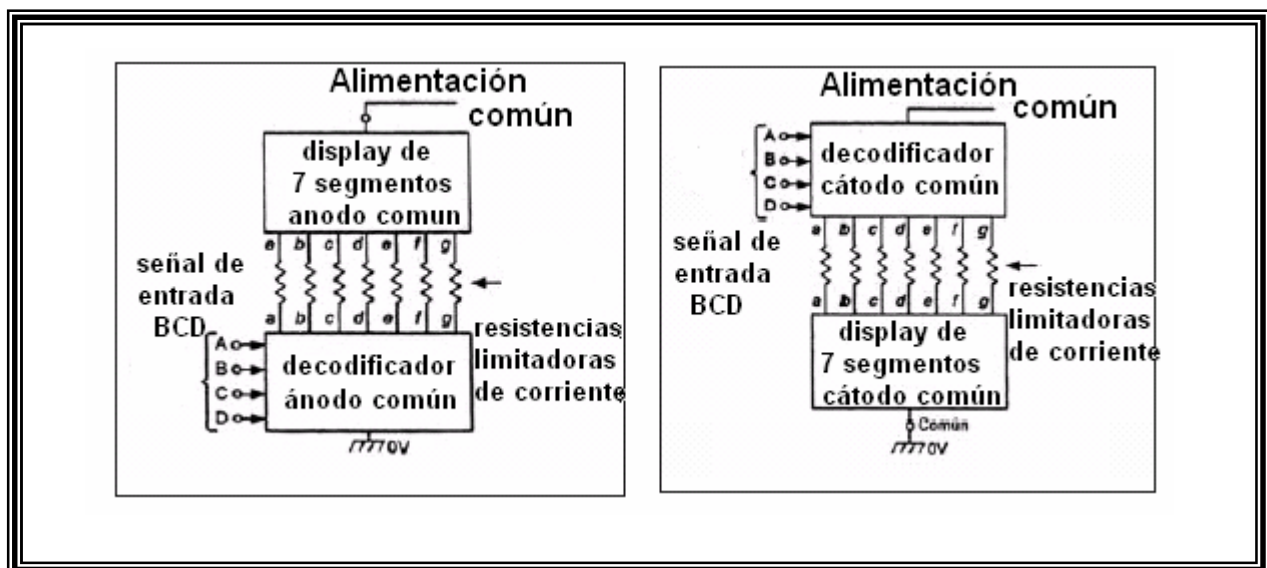


Figura 1.18. Display de ánodo y cátodo común ¹⁶

1.2.1. DISPLAY DE 7 SEGMENTOS LMS-4010BS ¹⁶

Este display tiene las dimensiones de 9x12cm de alto, se alimenta con 12V y es de ánodo común.



Figura 1.19. Display de ánodo común LMS-40101BS ¹⁶

1.2.2. CARACTERÍSTICAS ELÉCTRICAS DEL DISPLAY LMS-40101BS ⁷

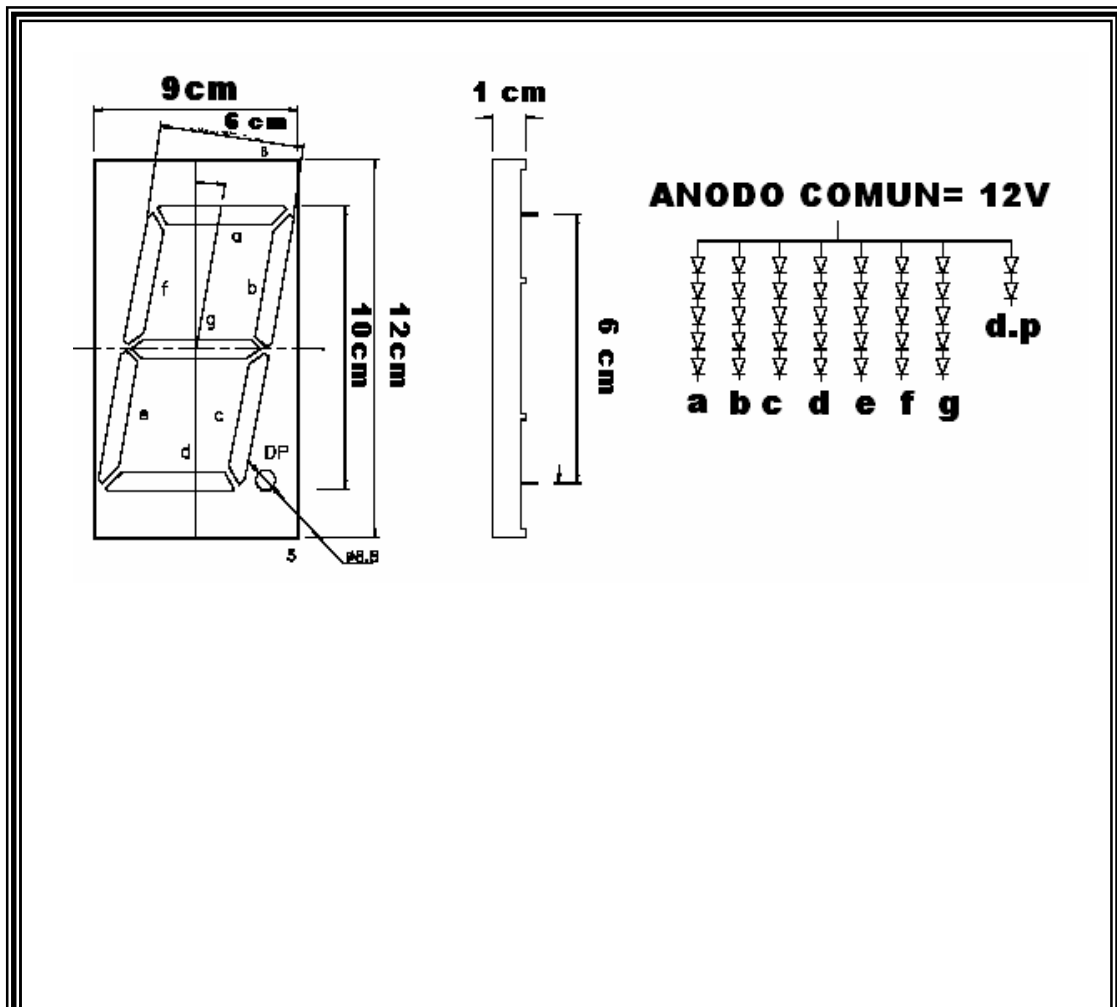




Figura 1.20. Configuración de pines del display LMS-40101BS⁷

⁷ <http://www.ucontrol.com.ar/Articulo20/Display4x7seg/4x7seg.htm>

Symbol	Parameter	Device	Typ.	Max.	Units	Test Conditions
λ_{peak}	Peak Wavelength	High Efficiency Red	627		nm	$I_F = 20mA$
λ_D	Dominate Wavelength	High Efficiency Red	625		nm	$I_F = 20mA$
$\Delta\lambda_{1/2}$	Spectral Line Half-width	High Efficiency Red	45		nm	$I_F = 20mA$
C	Capacitance	High Efficiency Red	15		pF	$V_F = 0V, f = 1MHz$
V_F	Forward Voltage	High Efficiency Red	7.5	10	V	$I_F = 20mA$
I_R	Reverse Current	High Efficiency Red		10	μA	$V_R = 5V$

Parameter	High Efficiency Red	Units
Power dissipation	105	mW
DC Forward Current	30	mA
Peak Forward Current [1]	160	mA
Reverse Voltage	12	V
Operating/Storage Temperature	-40°C To +85°C	
Lead Solder Temperature [2]	260°C For 5 Seconds	

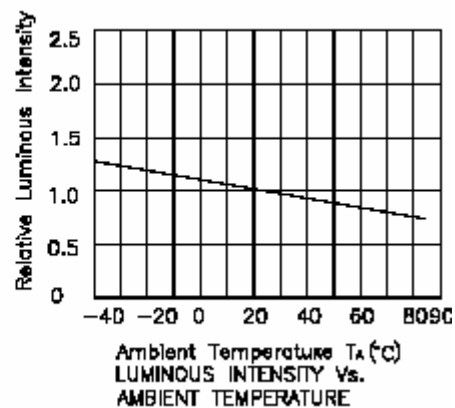
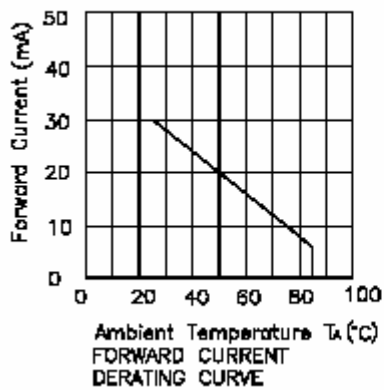
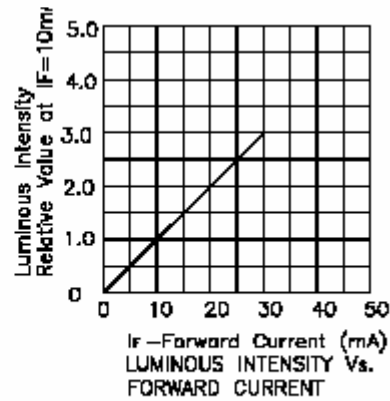
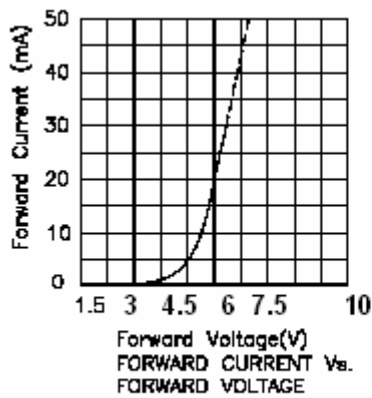


Figura 1.21. Características eléctricas del display LMS-40101BS⁷
 1.3. MANEJO DE DECODIFICADORES¹³

Un decodificador es lo contrario de un codificador, su función es la de transformar información binaria en información reconocible (números, letras, señales de aviso); se los clasifica en dos categorías: los lógicos y los controladores de visualizadores.

Los decodificadores lógicos se caracterizan por tener N entradas y M salidas, cada una de estas salidas responde a un único código de entrada binario.

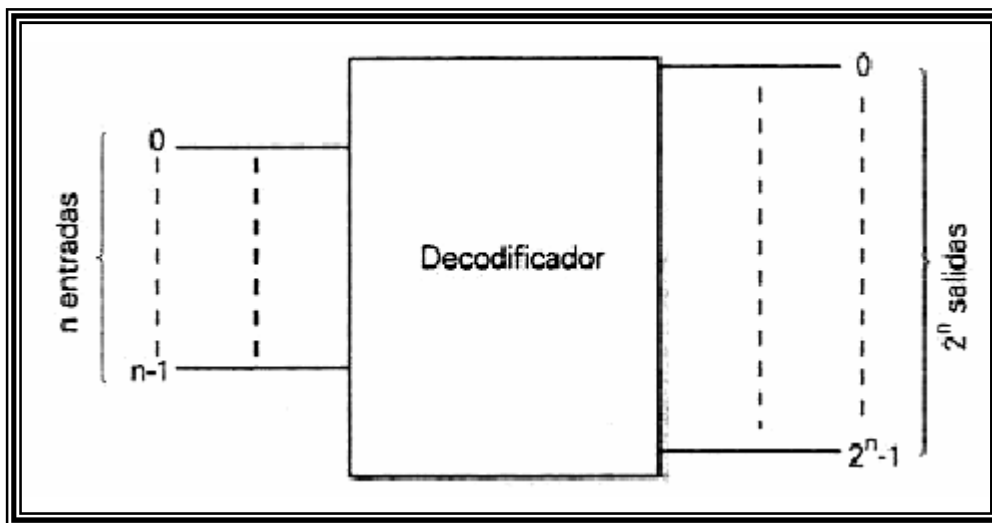


Figura 1.22. Decodificador lógico ¹³

Se puede tener decodificadores octales (3-8), BCD (4-10) y hexadecimales (4-16).

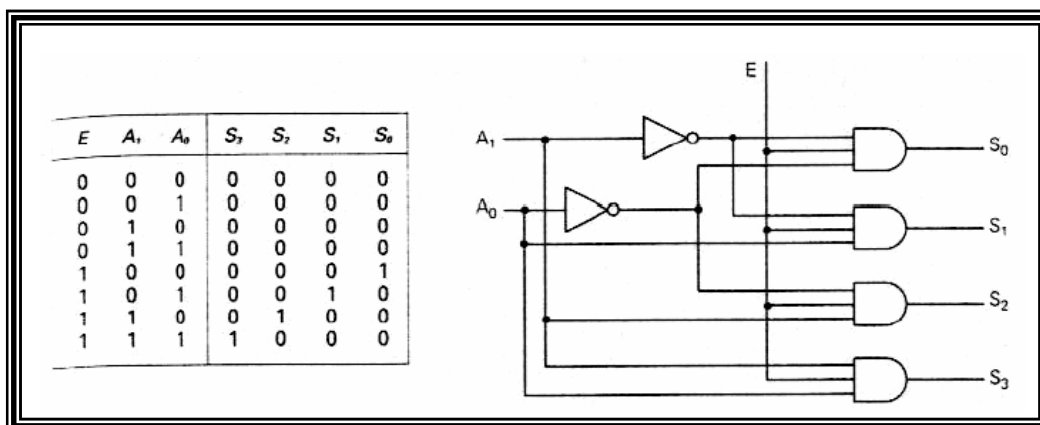


Figura 1.23. Diagrama lógico de un decodificador (2- 4) ¹³

¹³ <http://docs-europe.origin.electrocomponents.com/webdocs/020c/0900766b8020c616.pdf>

Este es un decodificador de 2 entradas y 4 salidas, el cual posee una entrada de habilitación; la entrada de habilitación E al hacerse cero coloca en cero todas las salidas sin importar el estado de las entradas A0 y A1.

1.3.1. DECODIFICADOR 7442 ¹³

Este decodificador es de Tecnología TTL con 4 entradas BCD y salidas decimales mutuamente exclusivas, las entradas son activas con nivel alto; mientras que las salidas se consideran activas con nivel bajo.

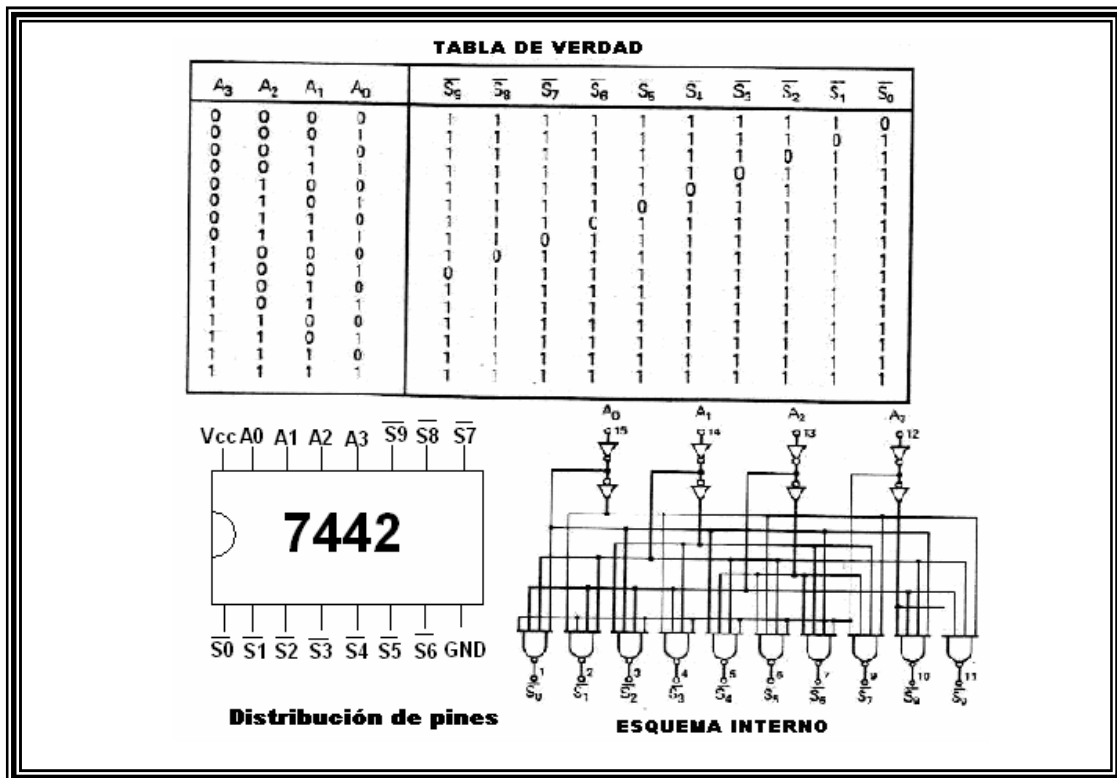


Figura 1.24. Decodificador 7442 ¹³

Los decodificadores controladores de visualizadores, son capaces de activar una o más salidas en respuesta a un código binario específico, por lo que se los utiliza para activar display, el más común es el decodificador de BCD a 7 segmentos; los cuales pueden ser de ánodo común o de cátodo común.

1.3.2. DECODIFICADOR 7447 ¹³

Este es uno de los decodificadores mas comúnmente utilizado para manejar displays de ánodo común de 7 segmentos.

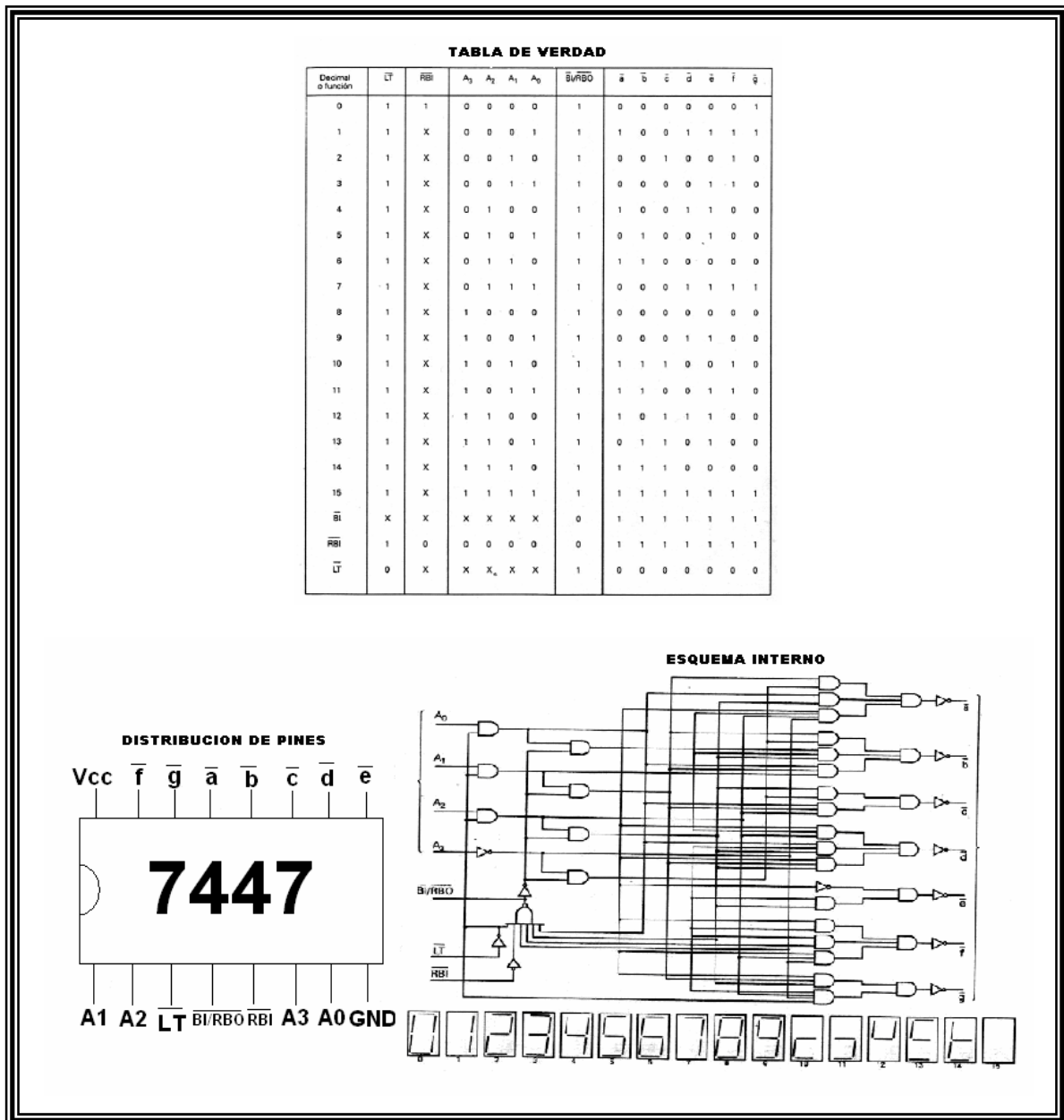


Figura 1.25. Decodificador 7447 ¹³

1.4. TRANSISTORES BIPOLARES, DARLINGTON NPN Y PNP ¹⁷

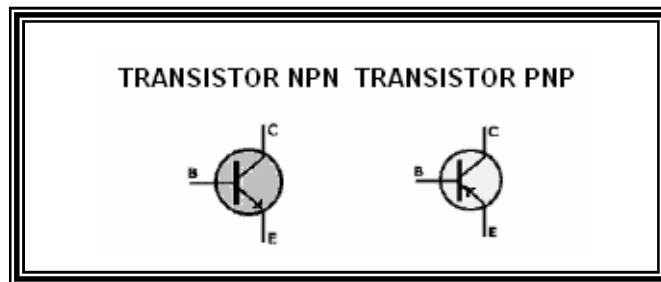


Figura 1.26. Transistor NPN y PNP ¹⁷

¹⁷ http://www.unicrom.com/Tut_transistor_bipolar.asp

El transistor bipolar es el más común de los transistores, y como los diodos, puede ser de germanio o silicio, existen dos tipos transistores: el NPN y el PNP, y la dirección del flujo de la corriente en cada caso, lo indica la flecha que corresponde a cada transistor. El transistor es un dispositivo de 3 terminales con los siguientes nombres: base (B), colector (C) y emisor (E).

El transistor es un amplificador de corriente, esto quiere decir que si le introducimos una cantidad de corriente por una de sus terminales (base), el entregará por otra (emisor), una cantidad mayor a ésta, en un factor que se llama amplificación. Este factor se llama β (beta) y es un dato propio de cada transistor.

Entonces:

I_c (corriente que pasa por el Terminal del colector) es igual a β (factor de amplificación) por I_b (corriente que pasa por el terminal de la base),
 $I_c = \beta * I_b$.

I_e (corriente que pasa por el Terminal del emisor) es del mismo valor que I_c , porque la corriente en un caso entra al transistor y en el otro caso sale de el, o viceversa. Según la fórmula anterior las corrientes no dependen del voltaje que alimenta el circuito (V_{cc}), pero en la realidad si lo hace y la corriente I_b cambia ligeramente cuando se cambia V_{cc} .

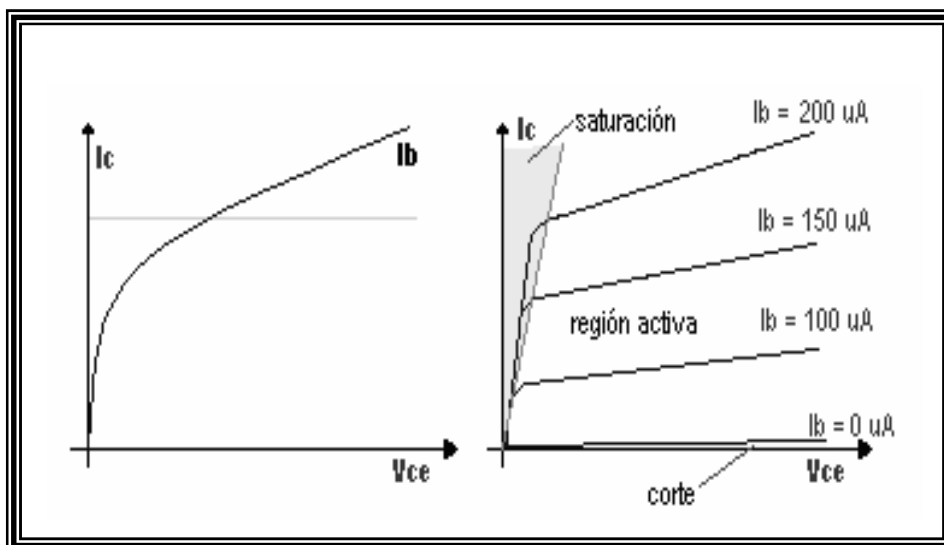


Figura 1.27. Región de corte, amplificación y saturación¹⁷

1.4.1. REGIONES OPERATIVAS DEL TRANSISTOR¹⁷

Región de corte: Un transistor está en corte cuando, Corriente de colector = corriente de emisor = 0, ($I_c = I_e = 0$); en este caso el voltaje entre el colector y el emisor del transistor es el voltaje de alimentación del circuito (como no hay corriente circulando, no hay caída de voltaje, según la ley de *Ohm*). Este caso normalmente se presenta cuando la corriente de base = 0 ($I_b = 0$).

Región de saturación: Un transistor está saturado cuando, Corriente de colector = corriente de emisor = corriente máxima, ($I_c = I_e = I_{\text{máxima}}$); en este caso la magnitud de la corriente depende del voltaje de alimentación del circuito y de las resistencias conectadas en el colector o el emisor o en ambos, *ver ley de Ohm*. Este caso normalmente se presenta cuando la corriente de base es lo suficientemente grande como para inducir una corriente de colector β veces más grande.

Región activa: Cuando un transistor no está ni en su región de saturación ni en la región de corte entonces está en una región intermedia, la región activa. En esta región la corriente de colector (I_c) depende principalmente de la corriente de base (I_b), de β (ganancia de corriente de un amplificador, es un dato del fabricante) y de las resistencias que hayan conectadas en el colector y emisor. Esta región es la más importante si desea utilizar al transistor como un amplificador.

Configuraciones: Hay tres tipos de configuraciones típicas en los amplificadores con transistores, cada una de ellas con características especiales que las hacen mejor para cierto tipo de aplicación: emisor común, colector común y base común; en el proyecto se ha utilizado el transistor NPN 2N3904, y el transistor PNP 2N3906, los cuales son complementarios; y tienen las mismas características de funcionamiento según su hoja de datos.

1.4.2. TRANSISTOR NPN 2N3904 Y PNP 2N3906¹⁷

Este transistor es muy utilizado como un switch electrónico por sus características tanto de corriente de colector I_c , ganancia, voltaje en el colector y su potencia de disipación.

NTE123AP
Silicon NPN Transistor
Audio Amplifier, Switch
(Compl to NTE159)

Absolute Maximum Ratings:

Collector–Emitter Voltage, V_{CE0}	40V
Collector–Base Voltage, V_{CB}	60V
Emitter–Base Voltage, V_{EB}	6V
Continuous Collector Current, I_C	600mA
Total Device Dissipation ($T_A = 25^\circ\text{C}$), P_D	350mW
Derate Above 25°C	2.8mW/ $^\circ\text{C}$
Total Device Dissipation ($T_C = 25^\circ\text{C}$), P_D	1.0W
Derate Above 25°C	8.0mW/ $^\circ\text{C}$
Operating Junction Temperature Range, T_J	-55° to $+150^\circ\text{C}$
Storage Temperature Range, T_{stg}	-55° to $+150^\circ\text{C}$
Thermal Resistance, Junction to Case, $R_{\theta JC}$	125 $^\circ\text{C/W}$
Thermal Resistance, Junction to Ambient, $R_{\theta JA}$	357 $^\circ\text{C/W}$

Electrical Characteristics: ($T_A = +25^\circ\text{C}$ unless otherwise specified)

Parameter	Symbol	Test Conditions	Min	Typ	Max	Unit
OFF Characteristics						
Collector–Emitter Breakdown Voltage	$V_{(BR)CEO}$	$I_C = 1\text{mA}$, $I_B = 0$, Note 1	40	–	–	V
Collector–Base Breakdown Voltage	$V_{(BR)CBO}$	$I_C = 0.1\text{mA}$, $I_E = 0$	60	–	–	V
Emitter–Base Breakdown Voltage	$V_{(BR)EBO}$	$I_E = 0.1\text{mA}$, $I_C = 0$	6	–	–	V
Collector Cutoff Current	I_{CEV}	$V_{CE} = 35\text{V}$, $V_{EB(off)} = 0.4\text{V}$	–	–	0.1	μA
Base Cutoff Current	I_{BEV}	$V_{CE} = 35\text{V}$, $V_{EB(off)} = 0.4\text{V}$	–	–	0.1	μA
ON Characteristics (Note 1)						
DC Current Gain	h_{FE}	$V_{CE} = 1\text{V}$, $I_C = 0.1\text{mA}$	20	–	–	
		$V_{CE} = 1\text{V}$, $I_C = 1\text{mA}$	40	–	–	
		$V_{CE} = 1\text{V}$, $I_C = 10\text{mA}$	80	–	–	
		$V_{CE} = 1\text{V}$, $I_C = 150\text{mA}$	100	–	300	
		$V_{CE} = 1\text{V}$, $I_C = 500\text{mA}$	40	–	–	

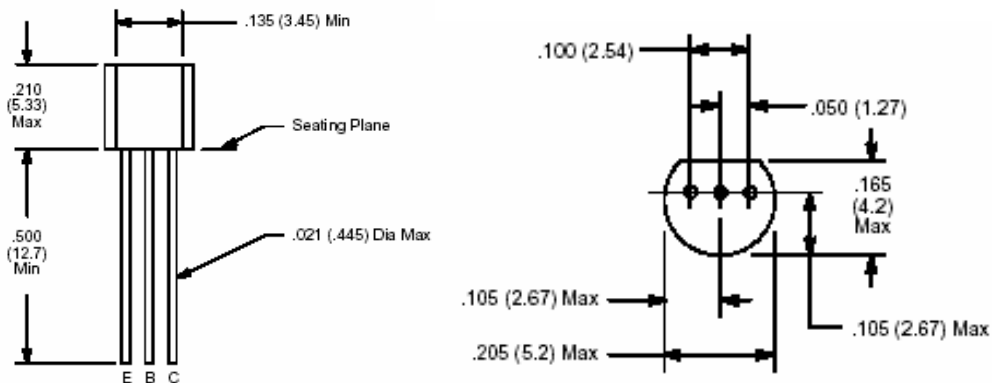


Figura 1.28. Características Eléctricas transistor NPN y PNP⁸

⁸ <http://www.ntepinc.com/specs/100to199/pdf/nte123ap.pdf>

1.4.3. TRANSISTOR DARLINGTON¹⁷

El transistor Darlington es un tipo especial de transistor que tiene una alta ganancia de corriente. Está compuesto internamente por dos transistores bipolares que se conectan en cascada; el transistor T1 entrega la corriente que sale por su emisor a la base del transistor T2. La ecuación de ganancia de un transistor es:

$I_E = \beta \times I_B$ (Corriente de colector es igual a beta por la corriente de base).

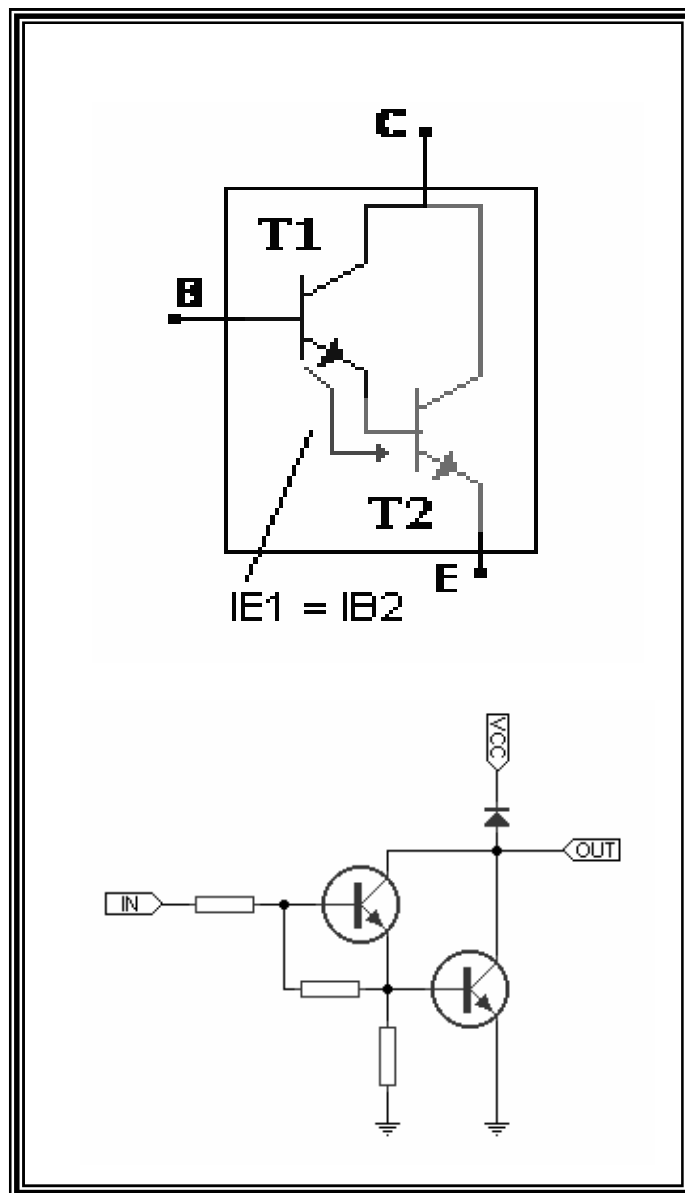


Figura 1.29. Transistor Darlington¹⁷

Ecuación del primer transistor es: $IE1 = \beta1 \times IB1$ (1), Ecuación del segundo transistor es: $IE2 = \beta2 \times IB2$ (2).

La corriente de emisor del transistor (T1) es la misma que la corriente de base del transistor T2. Entonces $IE1 = IB2$ (3)

Entonces utilizando la ecuación (2) y la ecuación (3) se obtiene:
 $IE2 = \beta2 \times IB2 = \beta2 \times IE1$

Reemplazando en la ecuación anterior el valor de $IE1$, se obtiene la ecuación final de ganancia del transistor Darlington.

$$IE2 = \beta2 \times \beta1 \times IB1$$

Este amplificador tiene una ganancia mucho mayor que la de un transistor normal, pues aprovecha la ganancia de los dos transistores (las ganancias se multiplican), por ejemplo: Si se tuvieran dos transistores con ganancia 100 ($\beta = 100$) conectados como un transistor Darlington y se utilizara la fórmula anterior, la ganancia sería, en teoría: $\beta2 \times \beta1 = 100 \times 100 = 10000$; La caída de tensión entre la base y el emisor del transistor Darlington es 1.4 voltios que resulta de la suma de las caídas de tensión de base a emisor del primer transistor B1 a E1 (0.7 voltios) y base a emisor del segundo transistor B2 y E2 (0.7 voltios).

Para nuestro caso se han usado los transistores darlington NPN TIP122, y el PNP TIP127, los cuales son complementarios y poseen las mismas características eléctricas. Además se utilizó el ECG268 el cual es del tipo NPN, con otra configuración de pines; los cuales se muestran a continuación según su hoja de datos. Todos estos transistores poseen internamente un diodo de protección en el colector, así como también dos resistencias en la base de cada transistor para realizar un divisor de tensión y polarizarlos correctamente.

Un dato importante que cabe destacar es que el voltaje de activación para cualquier transistor Darlington entre la base y el emisor tiene que ser de $V_{BE}=1.2V$ para que este correctamente polarizado.

1.4.4. TRANSISTORES DARLINGTON TIP122 Y TIP127⁹

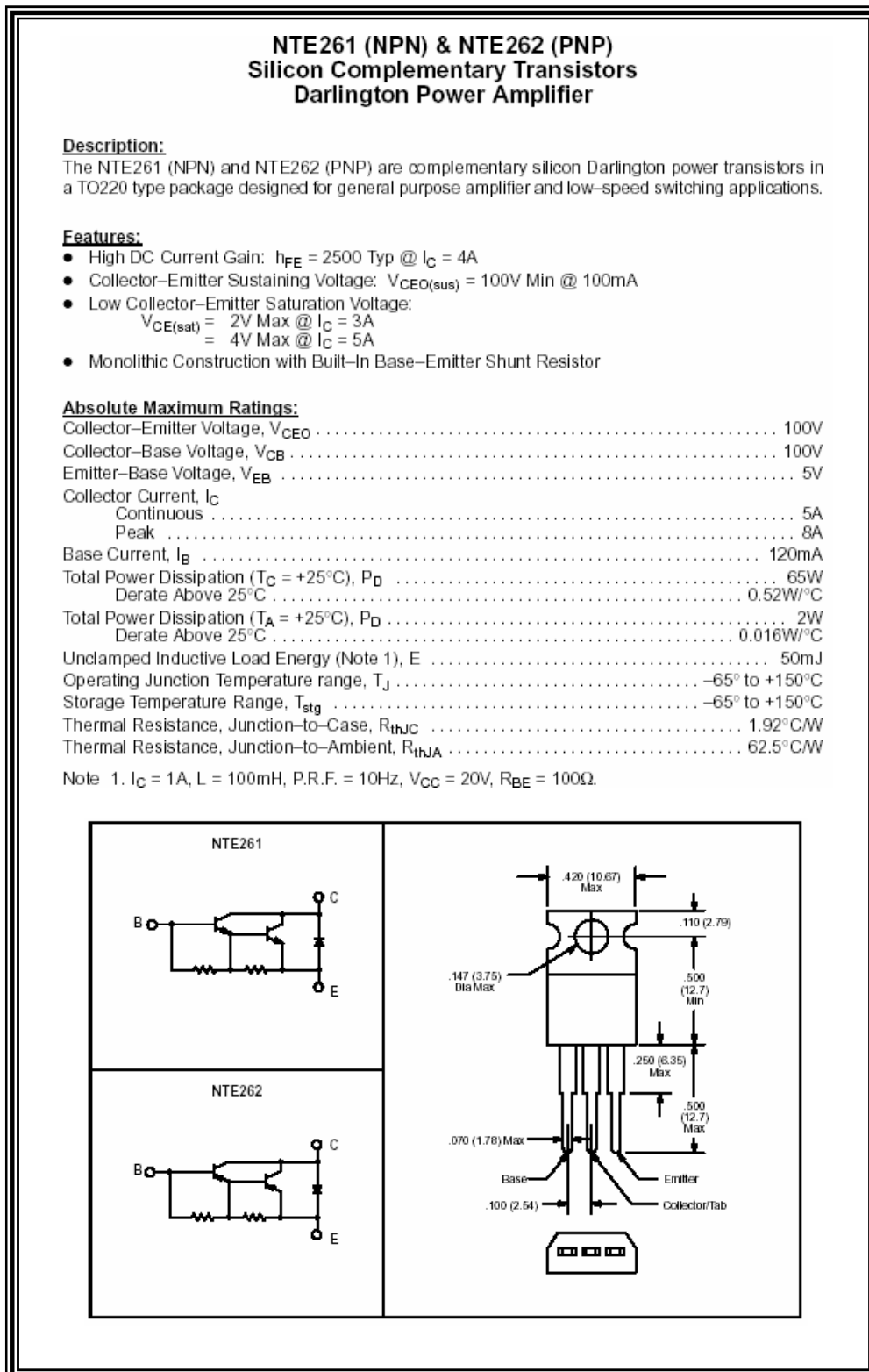


Figura 1.30. Características Eléctricas TIP122 Y TIP 127⁹

⁹ <http://www.ntepinc.com/specs/200to299/pdf/nte261.pdf>

1.4.5. TRANSISTOR DARLINGTON ECG 268 ¹⁰

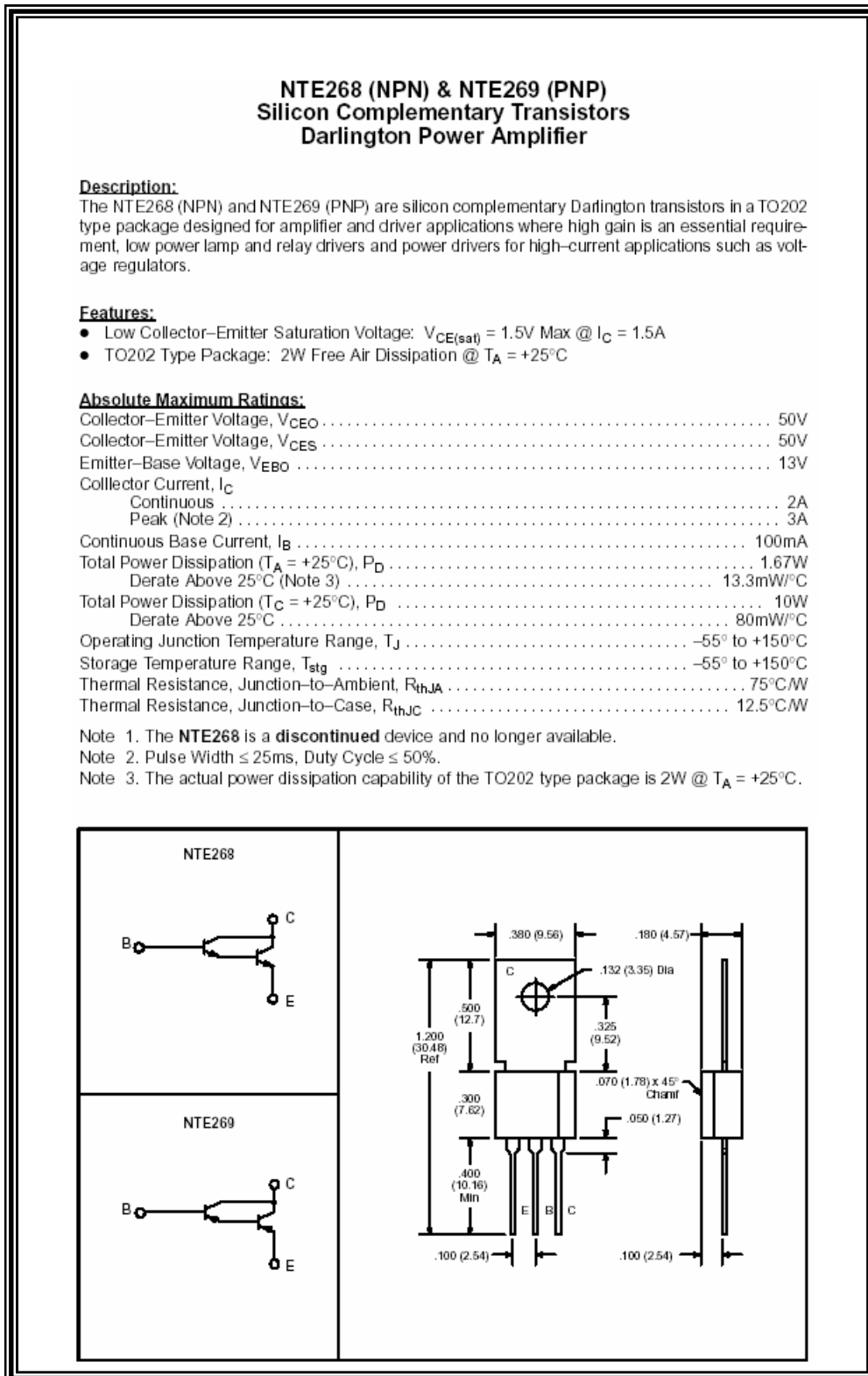


Figura 1.31. Características Eléctricas ECG 268 ¹⁰

¹⁰ <http://www.nteinc.com/specs/200to299/pdf/nte268.pdf>



CAPÍTULO II

2. DISEÑO Y CONSTRUCCIÓN DEL EQUIPO

2.1. IDENTIFICACIÓN DE LAS ETAPAS DEL CIRCUITO

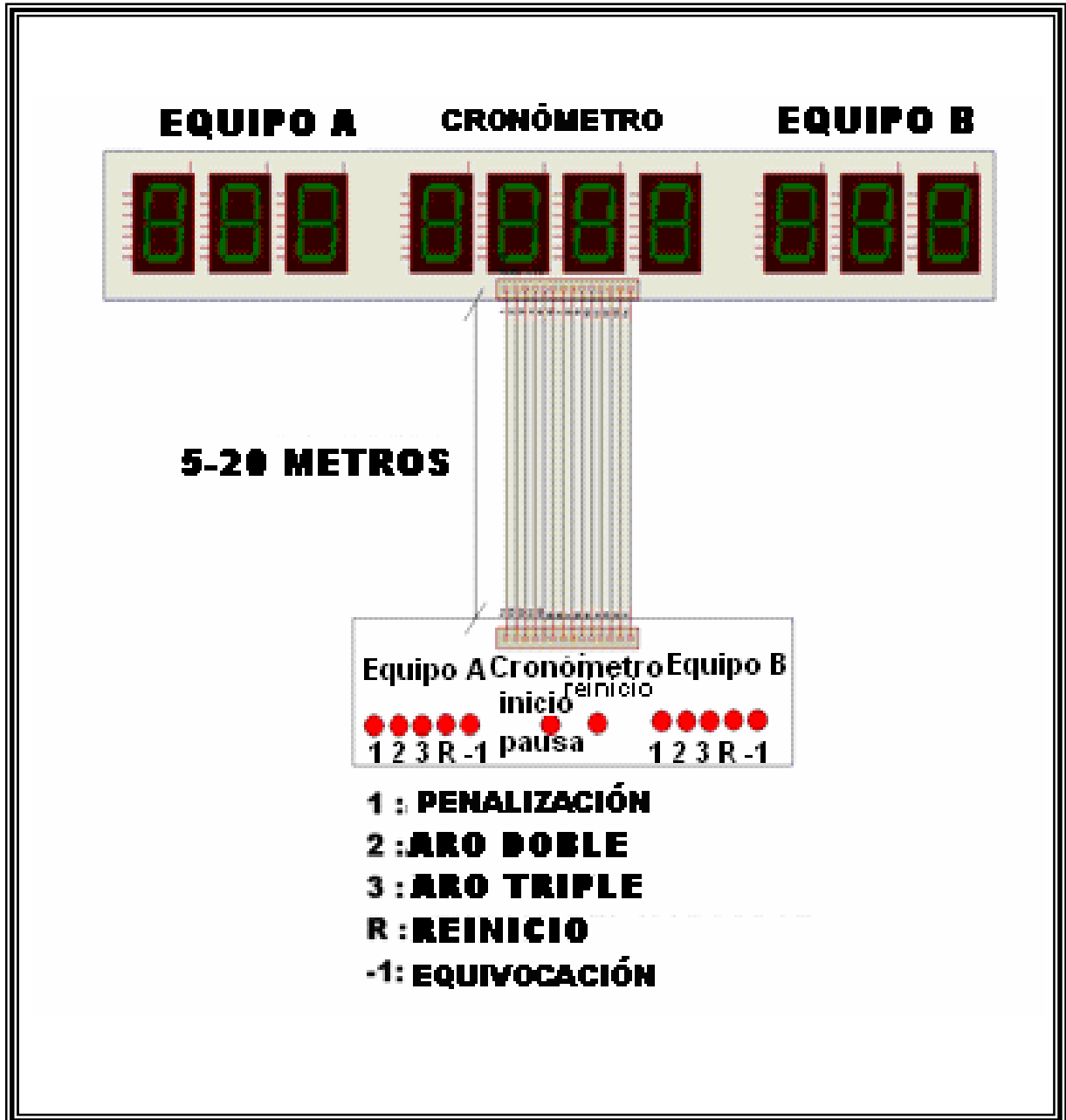


Figura 2.1. Identificación de las etapas del circuito

En la figura 2.1 se observa como quedó el equipo identificado con sus 3 etapas, la primera que es el contador del equipo A, el cronómetro de tiempo y el contador del equipo B.

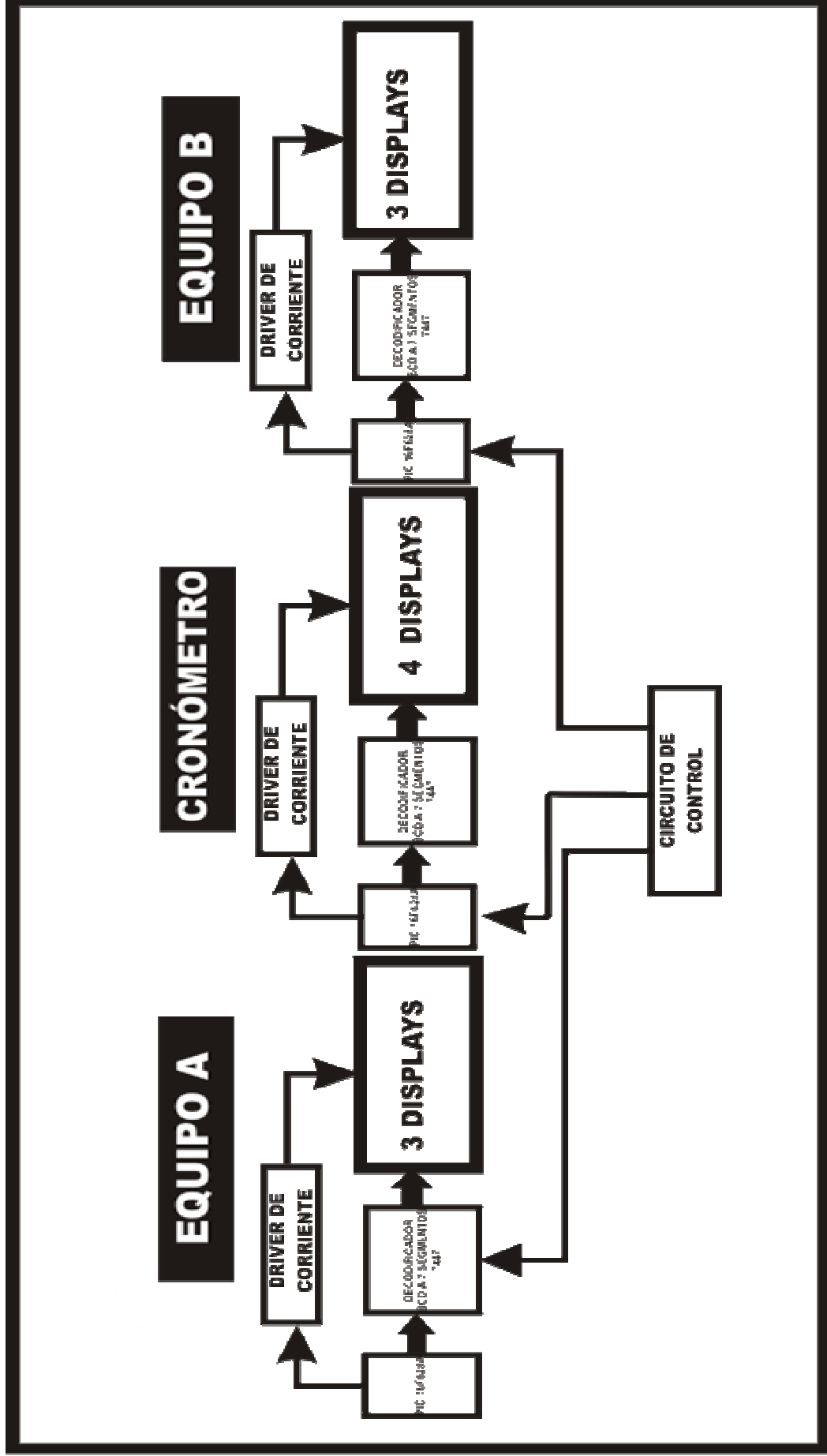


Figura 2.2. Diagrama Lógico del Circuito

En la figura 2.2 se observa que cada etapa del circuito está manejada por un PIC16F628A, en el caso del contador del equipo A posee 3 displays de ánodo común de 9x12cm para la visualización de los aros, además de un decodificador 7447 y 3 transistores Darlington PNP (TIP127) para la activación de cada display.

En el caso del contador del equipo B es relativamente idéntico al del equipo A, posee el mismo número de displays y elementos ya descritos para su correcto funcionamiento.

En cambio, en el cronómetro se va a utilizar 4 displays de ánodo común de tamaño similar al de los ya mencionados, los dos primeros serán para la visualización de los segundos y los otros dos para los minutos; 4 transistores Darlington PNP (TIP 127), y el decodificador 7447.

Además tanto el contador del equipo A y B, así como el cronómetro están conectados a una sirena de 12V/30W común; la cual se activa tanto para el incremento de aros, como para los tiempos de finalización y descanso del partido; con tiempos de duración fácilmente identificables.

2.2. DISEÑO DE LAS ETAPAS

Las etapas del circuito fueron diseñadas de la siguiente manera, se inicia describiendo a la parte principal del circuito la cual es el cronómetro.

2.2.1. CRONÓMETRO

En la figura 2.3 se observa que el cronómetro de tiempo está conformado por el PIC16F628A, este utiliza todo el puerto B0 como salidas de datos, tanto para sacar el código BCD hacia el decodificador 7447 y también para habilitar uno a uno los 4 transistores Darlington PNP TIP127, los cuales hacen a su vez que los 4 displays de ánodo común LMS-40011BS se enciendan al mismo tiempo, los dos primeros displays indican los segundos y los otros dos los minutos.

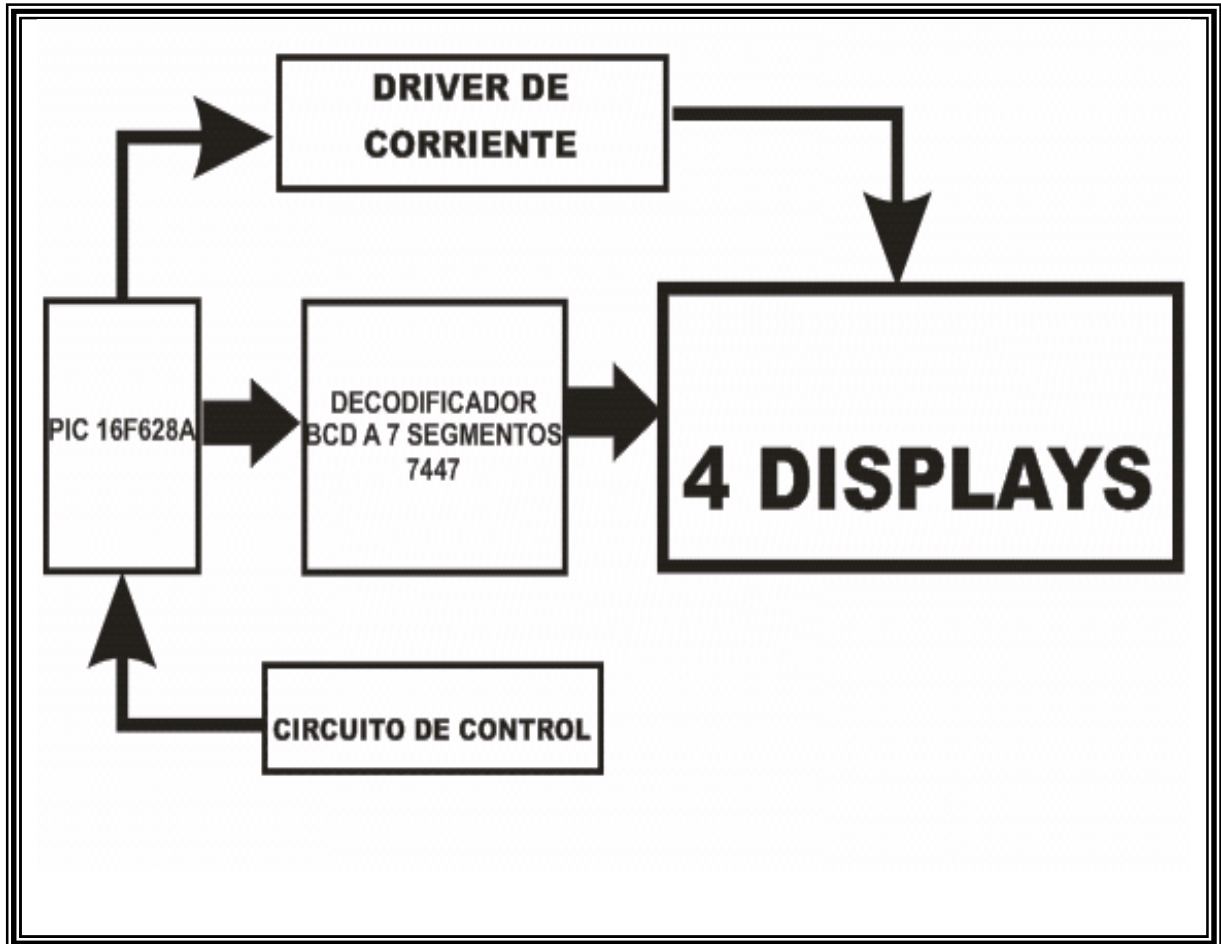


Figura 2.3. Diagrama lógico del Cronómetro

En la figura 2.4 se observa que se necesitan pines de control para poder dar el inicio y el reset del cronómetro, así como también un pin para habilitar a una sirena que indica los tiempos de juego; para esto se utilizan los 3 primeros pines del puerto A. Al mismo tiempo se debe tomar en cuenta que se necesitan dos voltajes de alimentación para esta etapa, uno de +5V, el otro de +12V y por supuesto tierra (GND) que es común para los dos voltajes requeridos.

Como se puede observar el diseño del cronómetro ha sido realizado, de tal forma que los displays de 7 segmentos de ánodo común, sean alimentados con un voltaje de 12V en su terminal común.

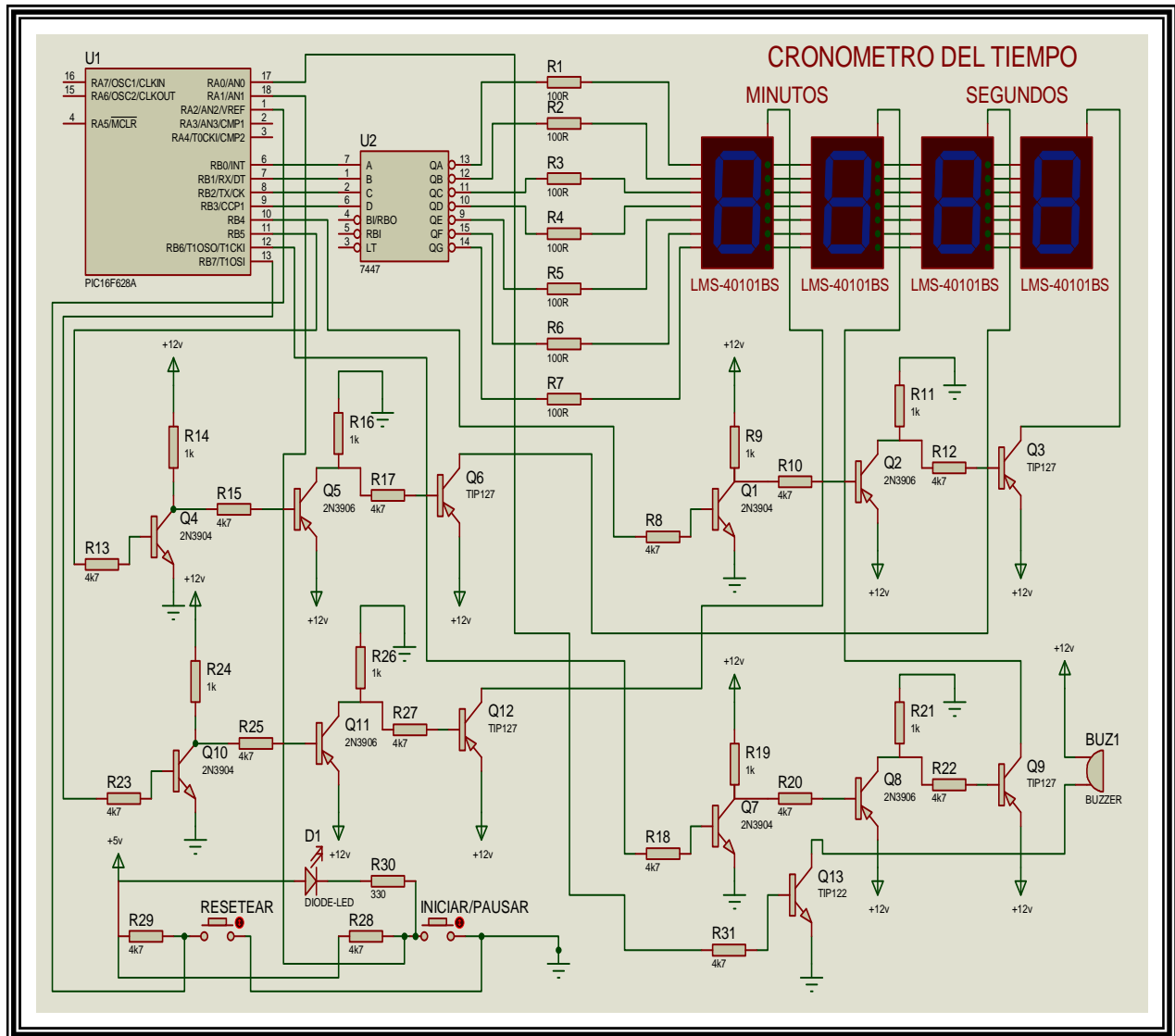


Figura 2.4. Diagrama esquemático del Cronómetro

2.2.1.1. Funcionamiento del Cronómetro

- El PIC activa a los transistores, con sus salidas en estado lógico bajo, o también conocido como 0V, o desactiva en estado lógico alto, o sea 5V.
- El transistor 2N3904 lo que hace es con 0V en su base, presentar a la salida, en su colector un nivel lógico alto, o sea 12V, ya que trabaja en la región de corte; para el otro caso cuando el PIC entregue 5V en su base; entrega en su colector un nivel lógico bajo, es decir 0V, en este caso el transistor está en la región de saturación, esto niveles lógicos los entrega al transistor 2N3906.

- El transistor 2N3906 con nivel lógico alto en su base (12V), presenta un nivel bajo en la salida (colector), y viceversa si se tiene (0v) en su base, entrega a la salida un nivel alto (12V), luego esto se lo entrega al transistor TIP127.

- Para el primer caso cuando esté un nivel lógico bajo en la base del transistor Darlington TIP127, entrega en el colector un nivel lógico alto de 12V, el cual es suficiente para lograr el encendido del display, en el otro caso con nivel lógico alto en la base del transistor ya mencionado, entrega 0V en el colector, por tanto el display se apaga.

- Todo este proceso que se realizó con transistores acoplados en cascada y se lo denomina, el aislamiento de la fuente. El objetivo es aislar los 5V que manda el PIC a través de sus pines; de los 12V que se tiene en el emisor del transistor Darlington TIP127, que nos sirve para la activación de cada display.

También se puede observar que posee 2 pulsadores para el control del tiempo, el primero está conectado a la línea 1 del puerto A, el cual es el pulsador del inicio o a su vez de pausa del partido, aquí cada vez que cuenta el cronómetro se enciende una luz piloto que indica que el cronómetro está funcionando correctamente y el segundo a la línea 2 del puerto A, este es el pulsador de reset del cronómetro. Así mismo se tiene que a la línea cero del puerto A está conectada la sirena la cual indica el momento en que arranca el partido, como también cuando se culminan los tiempos y descansos de juego reglamentarios.

Una vez realizado el diagrama esquemático del cronómetro y explicado el funcionamiento de cada elemento del circuito se procede a realizar la tarjeta de circuito impreso, para lo cual se utiliza el software Express PCB.

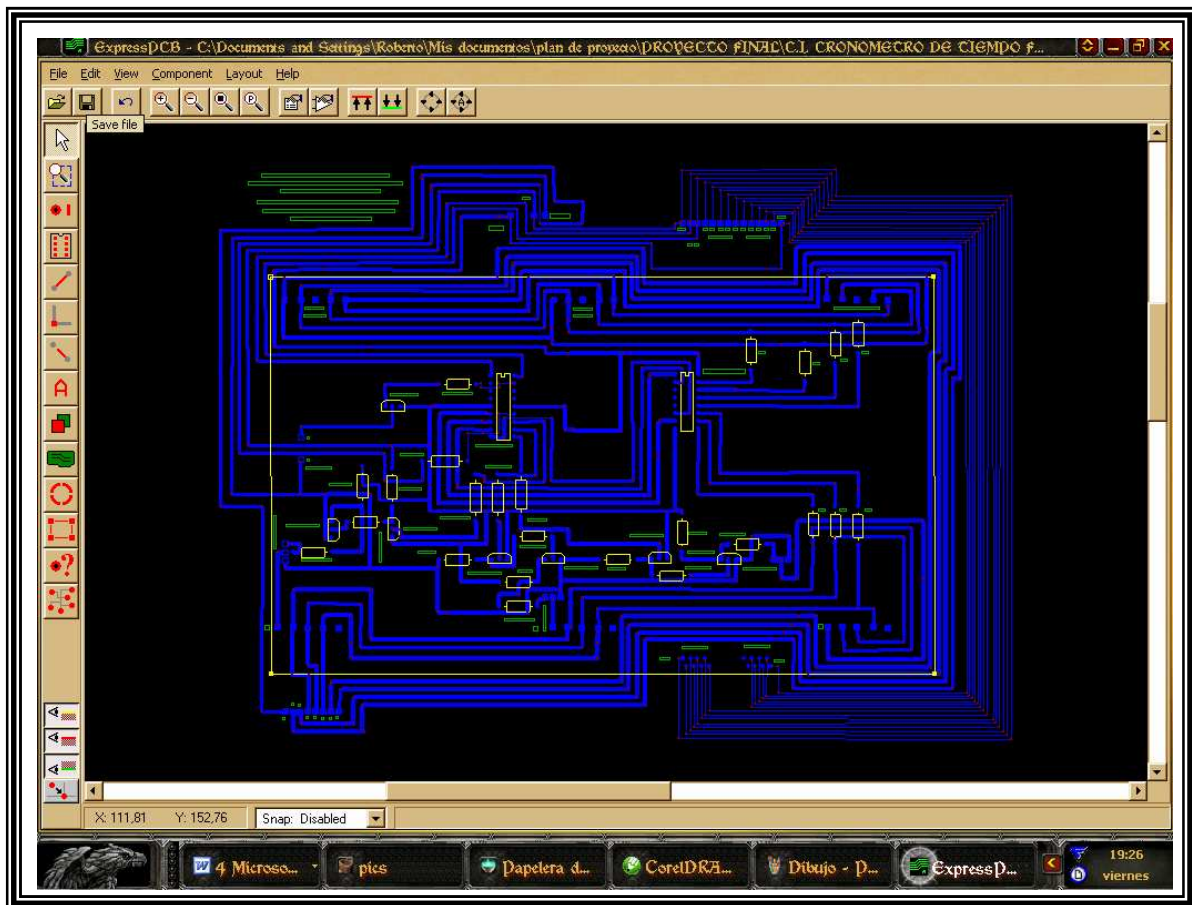


Figura 2.5. Express PCB

En esta tarjeta de circuito impreso consta lo siguiente: todos los elementos del diagrama esquemático del cronómetro, además tiene un conector de 10 terminales que sirve para establecer comunicación con las tarjetas del contador del equipo A y B, dos terminales RJ-45 que sirven para comunicarse con el circuito de control, también posee un conector de 8 terminales para comunicarse con el cuarto display que por motivos físicos se le tuvo que armar en una tarjeta adicional. Tiene un conector de 3 terminales para alimentarse de +5V, tierra y +12V.

También se coloca un terminal para la activación de la sirena.

2.2.2. CONTADOR DE AROS DEL EQUIPO A/B

En la figura 2.6 se observa que el contador de aros del equipo A/B está conformado por un PIC16F628A el cual, envía el código binario hacia el decodificador 7447 y la señal de activación hacia el driver de corriente, así

como también recibe la señal de incremento de aros del circuito de control; finalmente se representa el número indicado en los 3 displays.

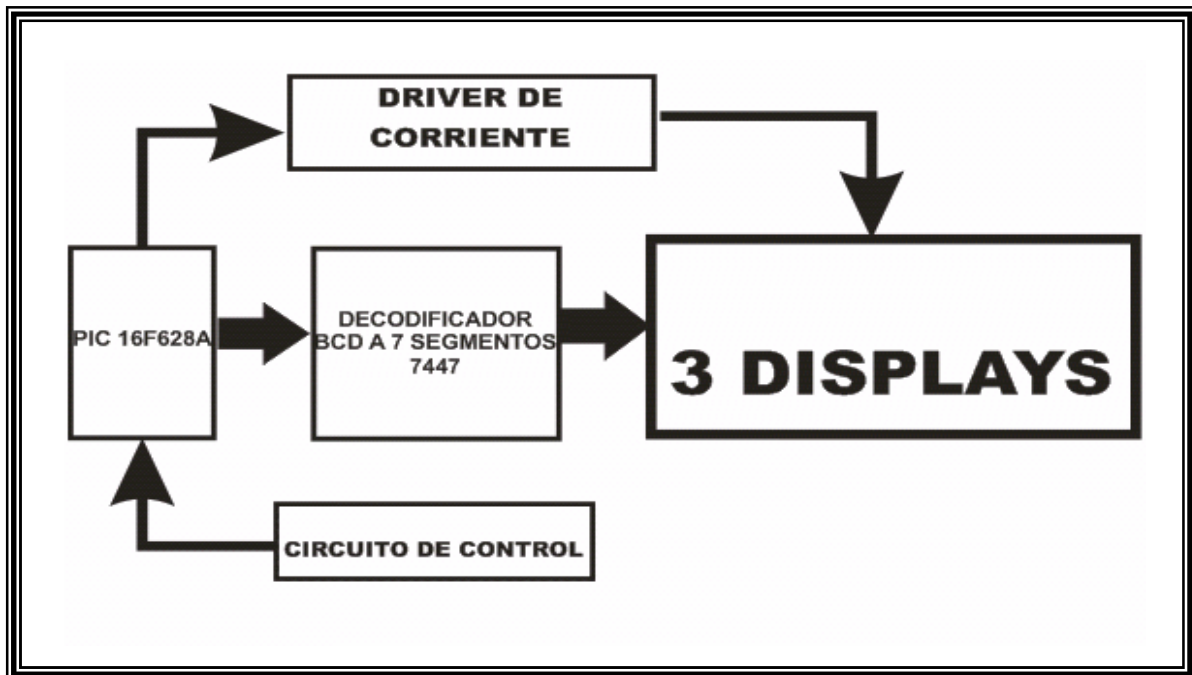


Figura 2.6. Diagrama Lógico del Contador de aros del Equipo A/B

2.2.2.1. Funcionamiento del Contador de aros del Equipo A/B

En el caso del contador de aros del Equipo A/B, el diseño se realiza de la siguiente manera, primero el PIC16F628A, a través de su puerto B0 con los bits menos significativos, es decir desde B0 hasta B3 envía los datos en BCD hacia el decodificador de BCD a 7 segmentos 7447; el cual luego envía los datos hacia los 4 displays. A continuación con los bits más significativos del puerto B0 desde el B4 hasta el B6, los cuales se utilizan para ir activando los displays, en este punto la activación de los displays es exactamente idéntica a la que se realizó para el cronómetro, utilizando primero el transistor 2N3904, luego el 2N3906 y finalmente el TIP127 para activar a cada uno de los 3 displays. Para el control del incremento de aros, se dispone del puerto A, de los cuales: El terminal A.0 es para la activación de la sirena de 12V/30W, el terminal A.1 para el incremento en 1 aro, el terminal A.2 para el incremento en 2 aros, el terminal A.3 para el incremento en 3 aros, el terminal A.4 para la disminución en 1 aro en el caso de la equivocación en el incremento de aros y el terminal A.6 para reiniciar el contador a cero.

Cabe destacar que todas las teclas ya descritas para el incremento, disminución o reinicio, son de antirrebote, esto quiere decir que por más que se le mantenga presionado el pulsador solo incrementa en el número correspondiente por una sola vez; todo esto para evitar errores en la contabilización de los aros.

Una vez explicado todo lo que se refiere al circuito esquemático del contador de aros del equipo A/B, se procede a realizar la tarjeta de circuito impreso, en la cual constan todos los elementos ya descritos; y además posee un conector de 5 terminales para la comunicación con el cronómetro. También tiene 1 conector de 3 terminales con alimentación de +5V, tierra y +12V, tal como se muestra en la figura 2.7.

Cabe mencionar que tiene un conector de 1 terminal para la activación de la sirena.

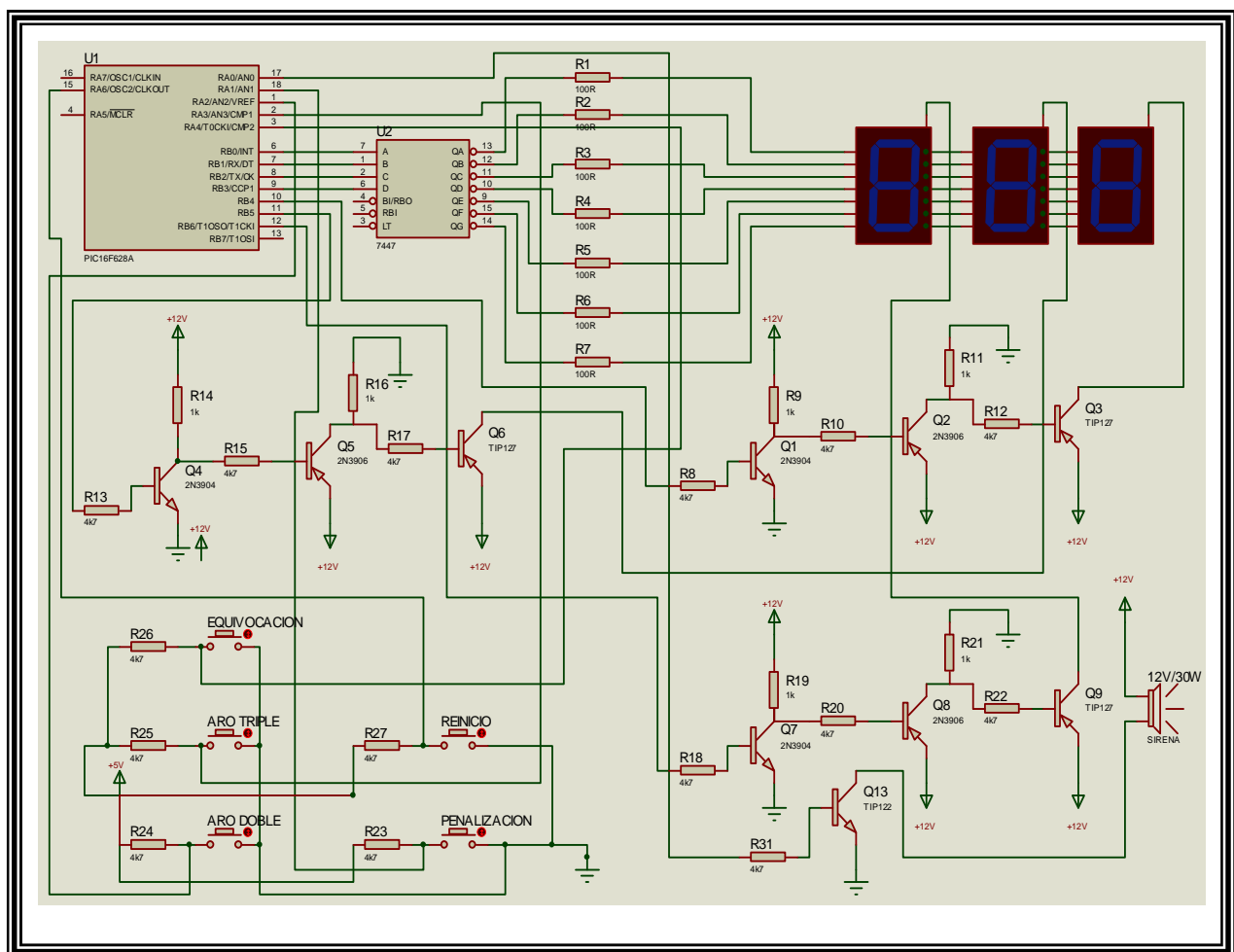


Figura 2.7. Contador de aros del Equipo A/B

2.2.3. CIRCUITO TOTAL

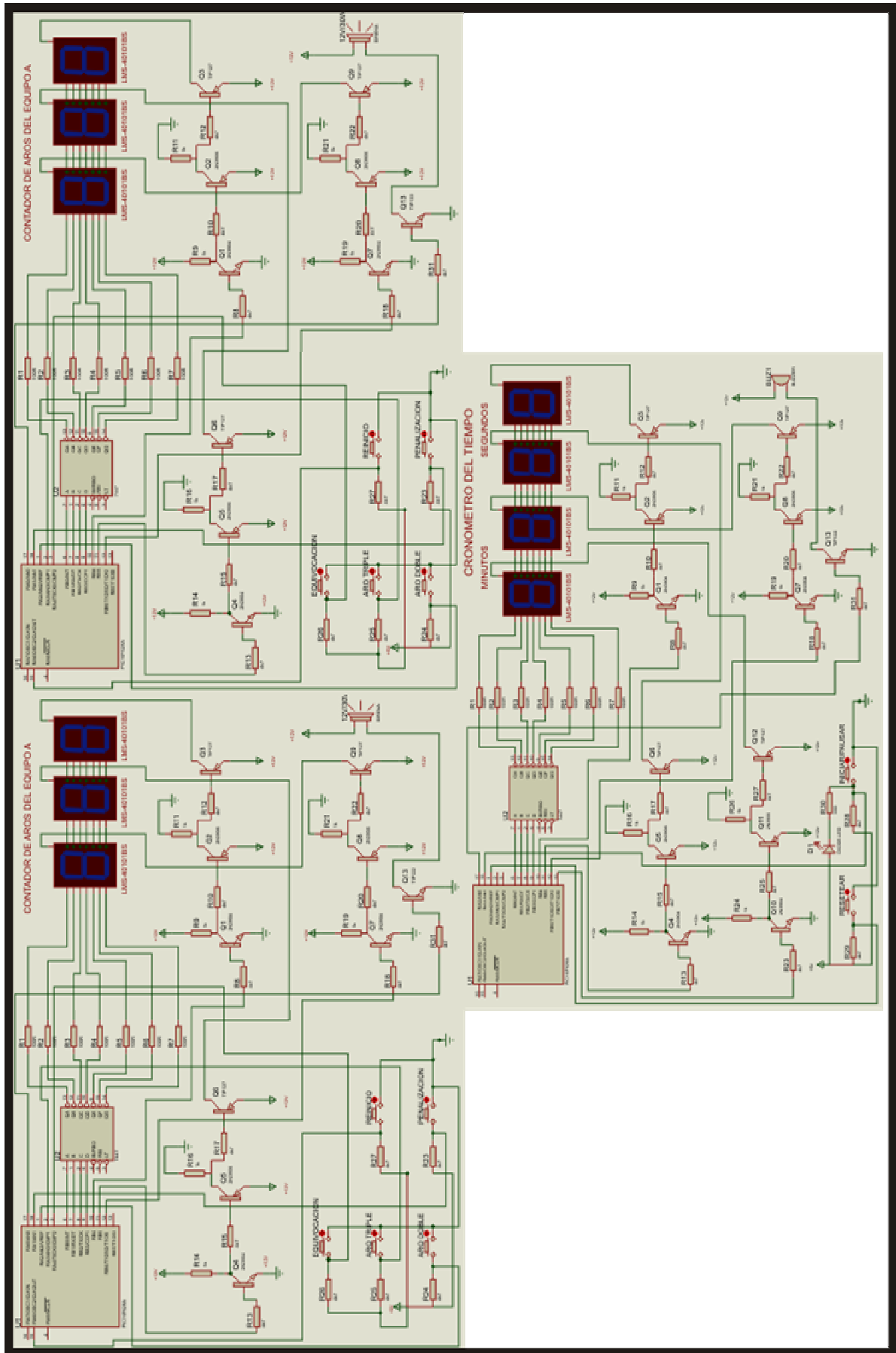


Figura 2.8. Circuito Total

2.3. PROGRAMACIÓN DEL MICROCONTROLADOR

Para la programación del microcontrolador PIC16F628A, se utiliza el siguiente software. Primero para elaborar el programa en lenguaje Basic se usa el programa llamado Microcode Studio Plus, el cual permite realizar el programa con el set de instrucciones ya descritas en el Capítulo I, este software permite compilar el programa y generar los archivos .ASM, .BAS y finalmente el .HEX, el cual es grabado en el PIC.

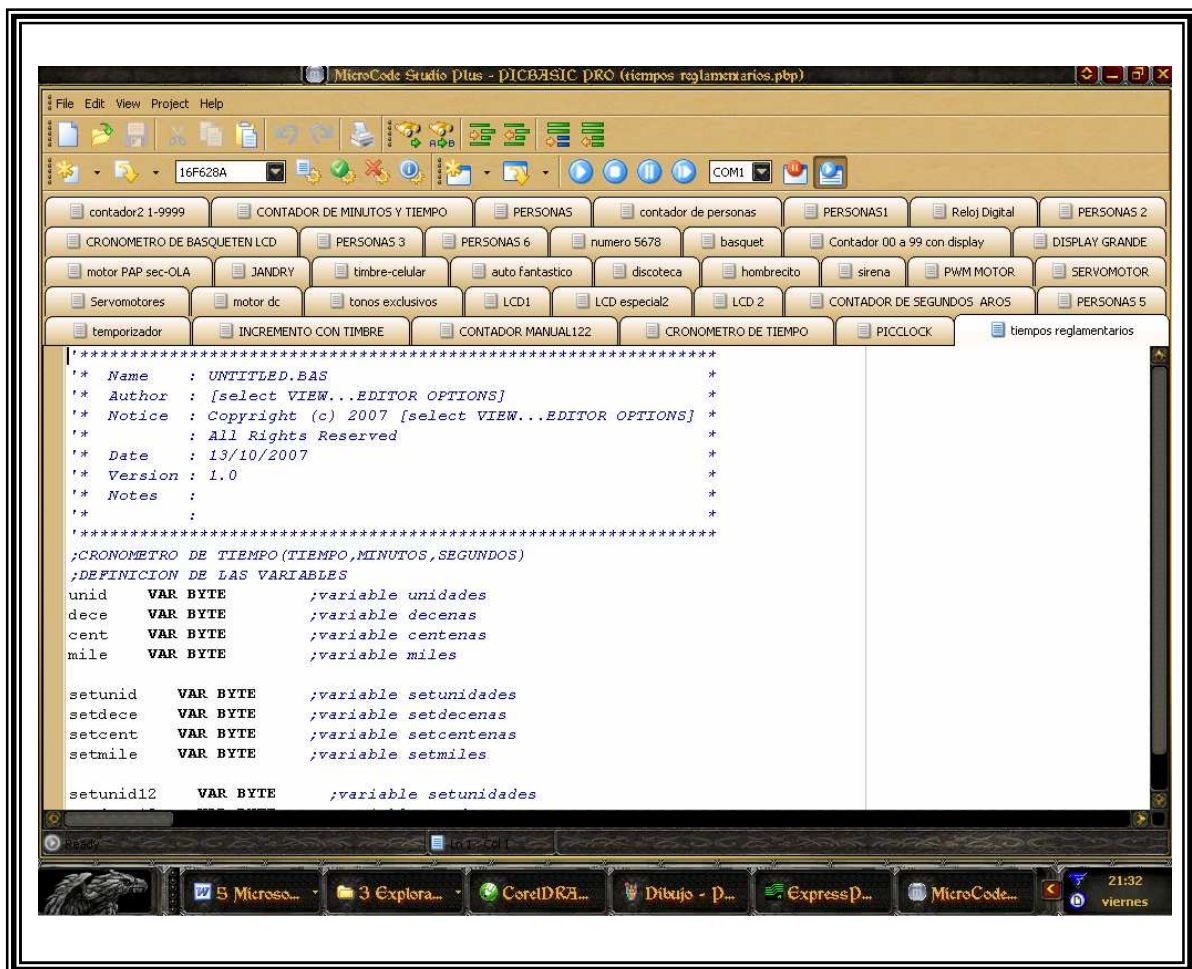


Figura 2.9. Microcode Studio Plus

Luego de elaborar el programa en este compilador, se necesita el software llamado WINPIC 3.55G, el cual es el encargado de grabar el archivo .HEX hacia el PIC.

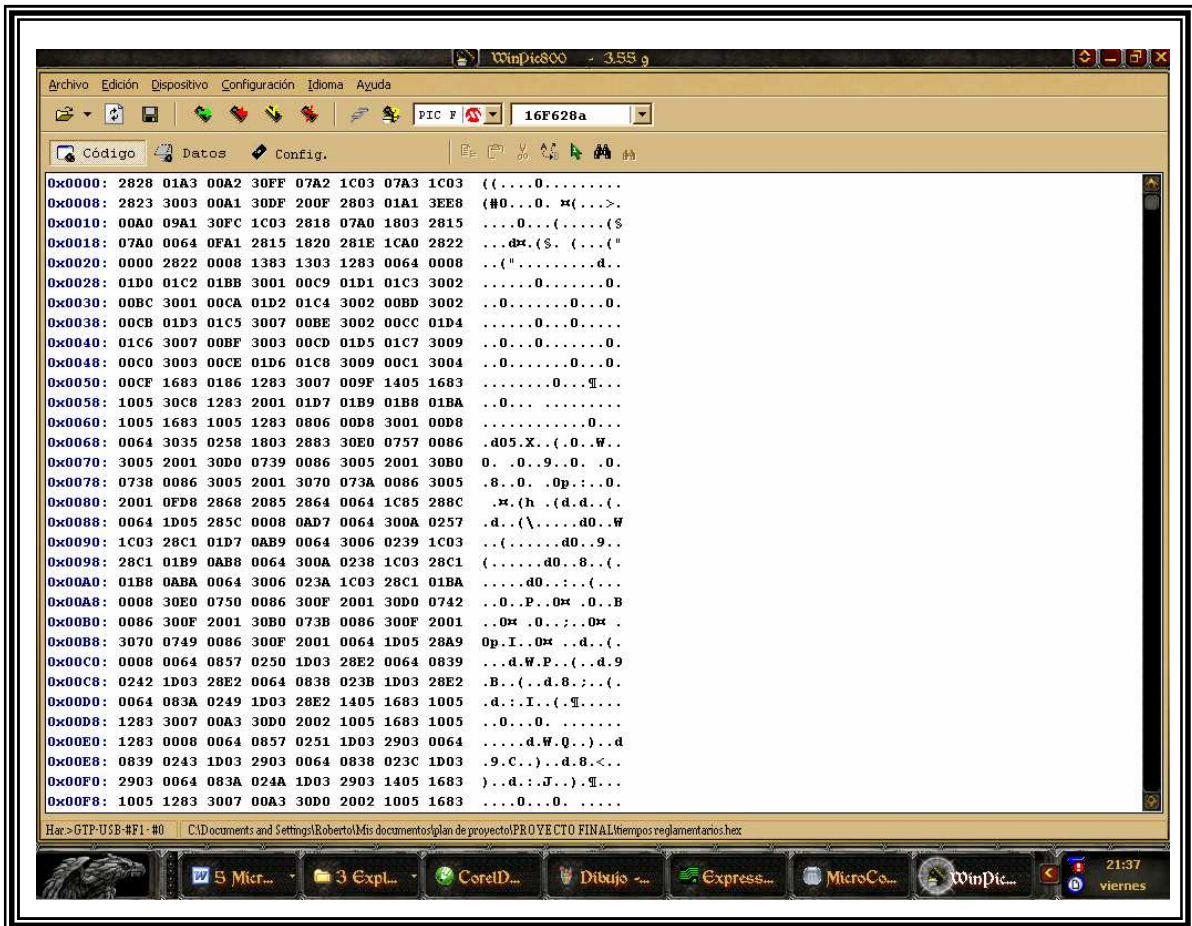


Figura 2.10. WIN PIC 3.55G

Finalmente para poder grabar este archivo hacia el PIC, se necesita establecer la comunicación entre la PC y el PIC, esto se lo realiza, a través del Programador de PICs, el cual funciona a través del puerto USB 2.0 y es de alta velocidad.

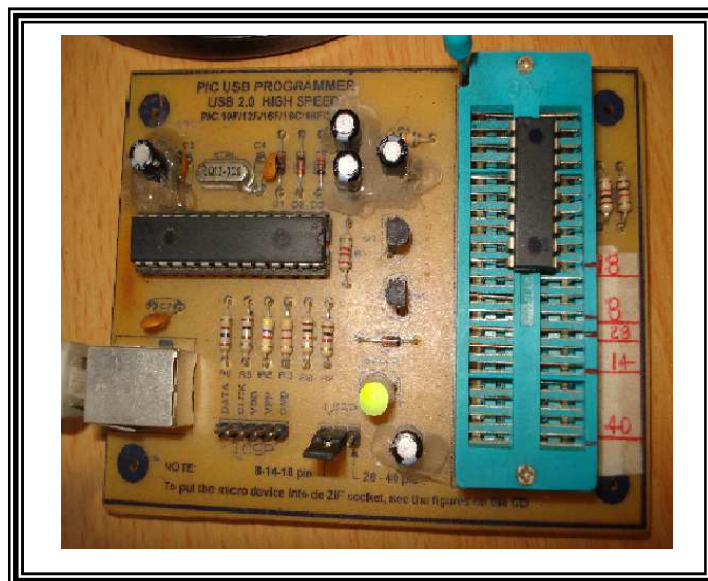


Figura 2.11. Programador USB 2.0

Los programas elaborados para el Cronómetro y contador de aros del Equipo A/B se los realiza con similares instrucciones y ligeras variaciones que se muestran a continuación.

2.3.1. DIAGRAMA DE FLUJO DEL CRONÓMETRO

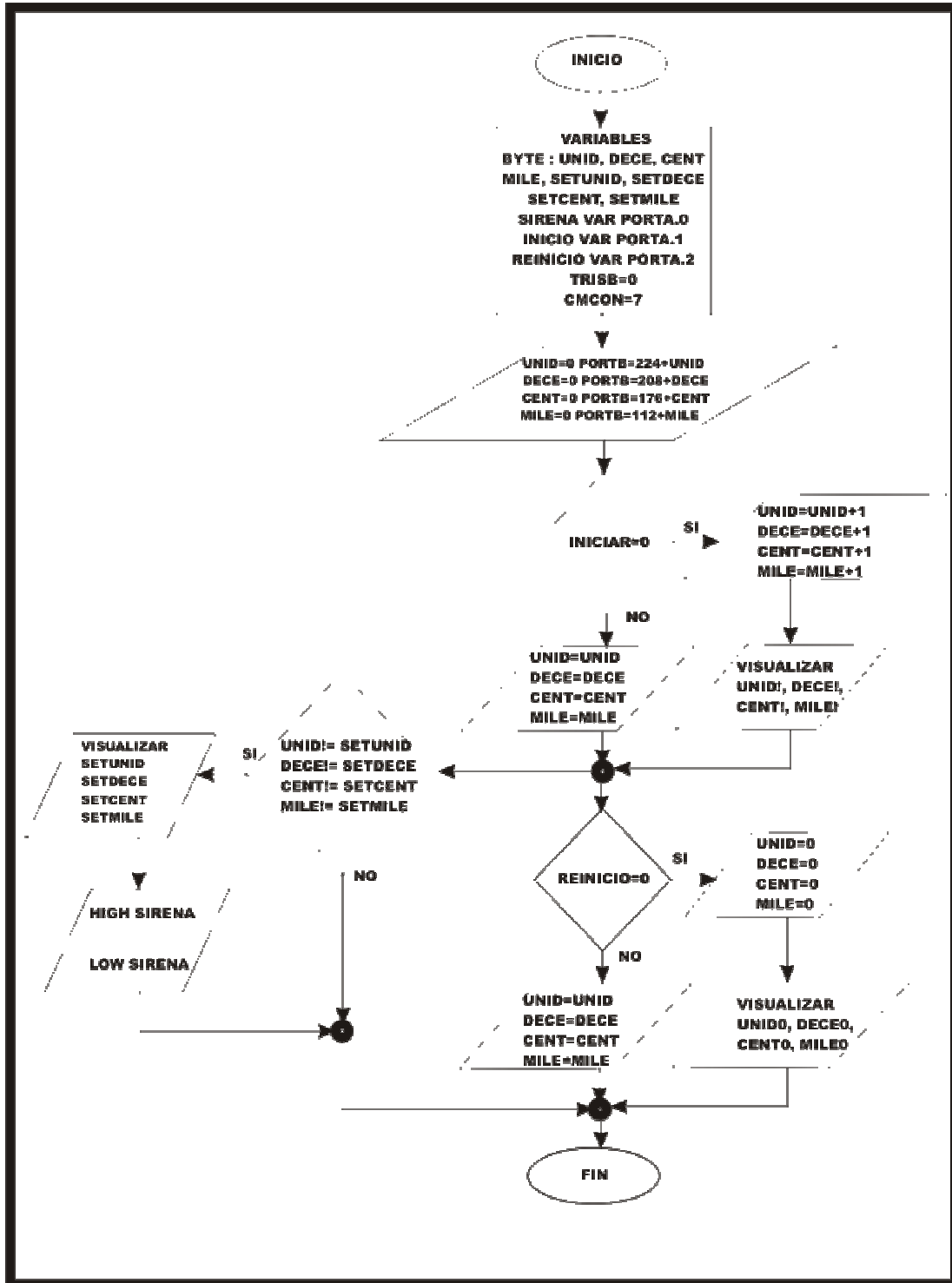


Figura 2.12. Diagrama de Flujo del Cronómetro

2.3.2. EXPLICACIÓN DEL PROGRAMA DEL CRONÓMETRO

En el diagrama de flujo se empieza definiendo las variables para almacenar el incremento de los números en unidad, decenas, centenas y miles con (unid, dece, cent, mile); así como también las variables de los tiempos de juego reglamentarios con (setunid, setdece, setcent, setmile), con su respectivo tiempo como son de 10, 12, 22, 27, 37, 39, 49 minutos. Luego se definen las variables para la utilización del puerto A, es así que la línea 0 es para la sirena, la línea 1 es para la tecla de inicio y la línea 2 para la tecla del reinicio del cronómetro.

En el desarrollo del programa se encera a las unidades, decenas centenas y miles, después se procede a su respectiva habilitación; se continua con condicionantes en el caso de que se cumpla la primera condición de (inicio=0) el contador incrementa en uno a las unidades, luego en uno a las decenas, en uno a las centenas y finalmente en uno a los miles. Este número formado por las (unid, dece, cent y mile) es almacenado en una variable y a su vez es comparado con el número almacenado al inicio del programa en (setunid, setdece, setcent y setmile), si estos números son iguales se activa la sirena como aviso de que se culminó el primero, segundo, tercero, último o el descanso respectivo; dependiendo del tiempo que este visible en el cronómetro.

En el caso de que no se cumpla la primera condición, se procede con la segunda condición que es (reinicio=0), si se cumple entonces se hace cero las unidades, decenas, centenas y miles; en el caso de que no se cumpla el cronómetro continua en pausa hasta que se habilite la tecla (inicio=0).

2.3.3. PROGRAMA DEL CRONÓMETRO

; Definición de las variables

unid	var byte	; variable byte de tamaño 255 unid
dece	var byte	; variable byte de tamaño 255 dece
cent	var byte	; variable byte de tamaño 255 cent
mile	var byte	; variable byte de tamaño 255 mile

; Variables para el primer período (10 minutos)

setunid var byte ; variable byte de tamaño 255 setunid
setdece var byte ; variable byte de tamaño 255 setdece
setcent var byte ; variable byte de tamaño 255 setcent
setmile var byte ; variable byte de tamaño 255 setmile

; Variables para el primer descanso (12 minutos)

setunid12 var byte ; variable byte de tamaño 255 setunid12
setdece12 var byte ; variable byte de tamaño 255 setdece12
setcent12 var byte ; variable byte de tamaño 255 setcent12
setmile12 var byte ; variable byte de tamaño 255 setmile12

; Variables para el segundo período (22 minutos)

setunid22 var byte ; variable byte de tamaño 255 setunid22
setdece22 var byte ; variable byte de tamaño 255 setdece22
setcent22 var byte ; variable byte de tamaño 255 setcent22
setmile22 var byte ; variable byte de tamaño 255 setmile22

; Variables para el segundo descanso (27 minutos)

setunid27 var byte ; variable byte de tamaño 255 setunid27
setdece27 var byte ; variable byte de tamaño 255 setdece27
setcent27 var byte ; variable byte de tamaño 255 setcent27
setmile27 var byte ; variable byte de tamaño 255 setmile27

; Variables para el tercer período (37 minutos)

setunid37 var byte ; variable byte de tamaño 255 setunid37
setdece37 var byte ; variable byte de tamaño 255 setdece37
setcent37 var byte ; variable byte de tamaño 255 setcent37
setmile37 var byte ; variable byte de tamaño 255 setmile37

; Variables para el tercer descanso (39 minutos)

setunid39 var byte ; variable byte de tamaño 255 setunid39
setdece39 var byte ; variable byte de tamaño 255 setdece39

setcent39 var byte ; variable byte de tamaño 255 setcent39
setmile39 var byte ; variable byte de tamaño 255 setmile39

; Variables para el cuarto período (49 minutos)

setunid49 var byte ; variable byte de tamaño 255 setunid49
setdece49 var byte ; variable byte de tamaño 255 setdece49
setcent49 var byte ; variable byte de tamaño 255 setcent49
setmile49 var byte ; variable byte de tamaño 255 setmile49
x var byte ; variable byte de tamaño 255 x

; Ingresamos los tiempos y descansos de juego reglamentarios

setunid =0 ; Primer período 1000
setdece=0
setcent =0
setmile =1

setunid12 =0 ; Primer Descanso 1200
setdece12 =0
setcent12 =2
setmile12 =1

setunid22 =0 ; Segundo período 2200
setdece22 =0
setcent22 =2
setmile22 =2

setunid27 =0 ; Segundo Descanso 2700
setdece27 =0
setcent27 =7
setmile27 =2

setunid37 =0 ; Tercer período 3700
setdece37 =0
setcent37 =7

setmile37 =3

setunid39 =0 ; Tercer Descanso 3900

setdece39 =0

setcent39 =9

setmile39 =3

setunid49 =0 ; Cuarto período 4900

setdece49 =0

setcent49 =9

setmile49 =4

; Definición De Variables Para Las Líneas Del Puerto A

Sirena var porta.0 ; Terminal De La Sirena

Inicio var porta.1 ; Tecla Inicio/Pausa

Reinicio var porta.2 ; Tecla De Reinicio

Trisb =0 ; Puerto B Como Salida

Cmcon =7 ; Puerto A En Modo Digital

high Sirena ; Se Activa La Sirena

pause 200

; Desarrollo del programa

encerrar:

unid =0 ; carga la variable unid con cero

dece =0 ; carga la variable dece con cero

cent =0 ; carga la variable cent con cero

mile =0 ; carga la variable mile con cero

low Sirena ; apagar la Sirena

Display:

x=portb

for x=1 to 52

portb= 224+unid ;224 %11100000,activa las unidades

pause 5

```
portb= 208+dece ;208 %11010000,activa las decenas
pause 5
portb= 176+cent ;176 %10110000,activa las centenas
pause 5
portb= 112+mile ;112 %01110000,activa los miles
pause 5
next
gosub teclas
goto display
```

teclas:

```
if iniciar =0 then contador
if encera =0 then encerar
return
```

contador:

```
unid=unid+1 ;suma 1 a las unidades
if unid<10 then comparar ;si unid es menor a 10 comparar
unid=0 ;hace cero a las unidades
dece=dece+1 ;y incrementa las decenas
if dece<6 then comparar
dece=0
cent=cent+1
if cent<10 then comparar
cent=0
mile=mile+1
if mile<6 then comparar
mile=0
return
```

Visualizar:

```
portb= 224+setunid ;224 %11100000,activa las unidades
pause 15
portb= 208+setdece ;208 %11010000,activa las decenas
```



```
pause 15
portb= 176+setcent    ;176 %01110000,activa las centenas
pause 15
portb= 112+setmile    ;112 %01110000,activa los miles
pause 15
if encera=0          then visualizar
return                ; retornar a gosub teclas
```

; Compara el valor que se tiene almacenado en (unid!, dec!, cent!, mile!) con 10

Comparar10:

```
if unid!=setunid then comparar12 ;si unid no es igual a setunid
if dece!=setdece then comparar12 ;si dece no es igual a setdece
if cent!=setcent then comparar12
if mile!=setmile then comparar12
high sirena                ;encender la sirena
pause 2000
low sirena                 ;apagar la sirena
return                     ;retornar a gosub teclas
```

; Compara el valor que se tiene almacenado en (unid!, dec!, cent!, mile!) con 12

Comparar 12:

```
if unid!=setunid12 then comparar22
if dece!=setdece12 then comparar22
if cent!=setcent12 then comparar22
if mile!=setmile12 then comparar22
high sirena
pause 2000
low sirena
return
```

; Compara el valor que se tiene almacenado en (unid!, dec!, cent!, mile!) con 22

Comparar 22:

```
if unid!=setunid22 then comparar27
if dece!=setdece22 then comparar27
```

```
if cent!=setcent22 then comparar27
if mile!=setmile22 then comparar27
high sirena
pause 2000
low sirena
return
```

; Compara el valor que se tiene almacenado en (unid!, dec!, cent!, mile!) con 27

Comparar 27:

```
if unid!=setunid27 then comparar37
if dece!=setdece27 then comparar37
if cent!=setcent27 then comparar37
if mile!=setmile27 then comparar37
high sirena
pause 2000
low sirena
return
```

; Compara el valor que se tiene almacenado en (unid!, dec!, cent!, mile!) con 37

Comparar 37:

```
if unid!=setunid37 then comparar39
if dece!=setdece37 then comparar39
if cent!=setcent37 then comparar39
if mile!=setmile37 then comparar39
high sirena
pause 2000
low sirena
return
```

; Compara el valor que se tiene almacenado en (unid!, dec!, cent!, mile!) con 39

Comparar 39:

```
if unid!=setunid39 then comparar49
if dece!=setdece39 then comparar49
if cent!=setcent39 then comparar49
```

```
if mile!=setmile39 then comparar49
high sirena
pause 2000
low sirena
return
```

; Compara el valor que se tiene almacenado en (unid!, dec!, cent!, mile!) con 49
Comparar 49:

```
if unid!=setunid49 then return
if dece!=setdece49 then return
if cent!=setcent49 then return
if mile!=setmile49 then return
high sirena
pause 3000
low sirena
return
End
```

Nota: Para la programación del cronómetro se ha toma en cuenta lo siguiente, que en el basket amateur que se juega en nuestro medio, los tiempos están distribuidos en 4 de 10 minutos cada uno, con intervalos de descanso; después del primer tiempo o más conocido como primer período existe 2 minutos de descanso, después del segundo período hay 5 minutos de descanso, y después del tercer período 2 minutos más de descanso y finalmente finaliza el partido en el cuarto período, Ejemplo:

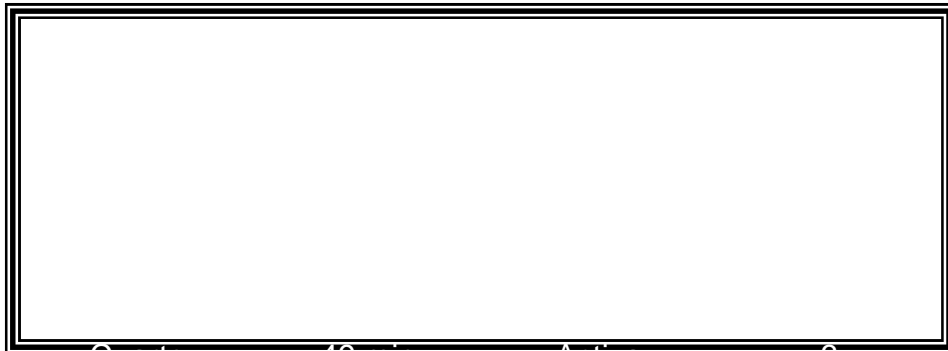


Figura 2.13. Programación del Cronómetro

2.3.4. DIAGRAMA DE FLUJO DEL CONTADOR DE AROS DEL EQUIPO A/B

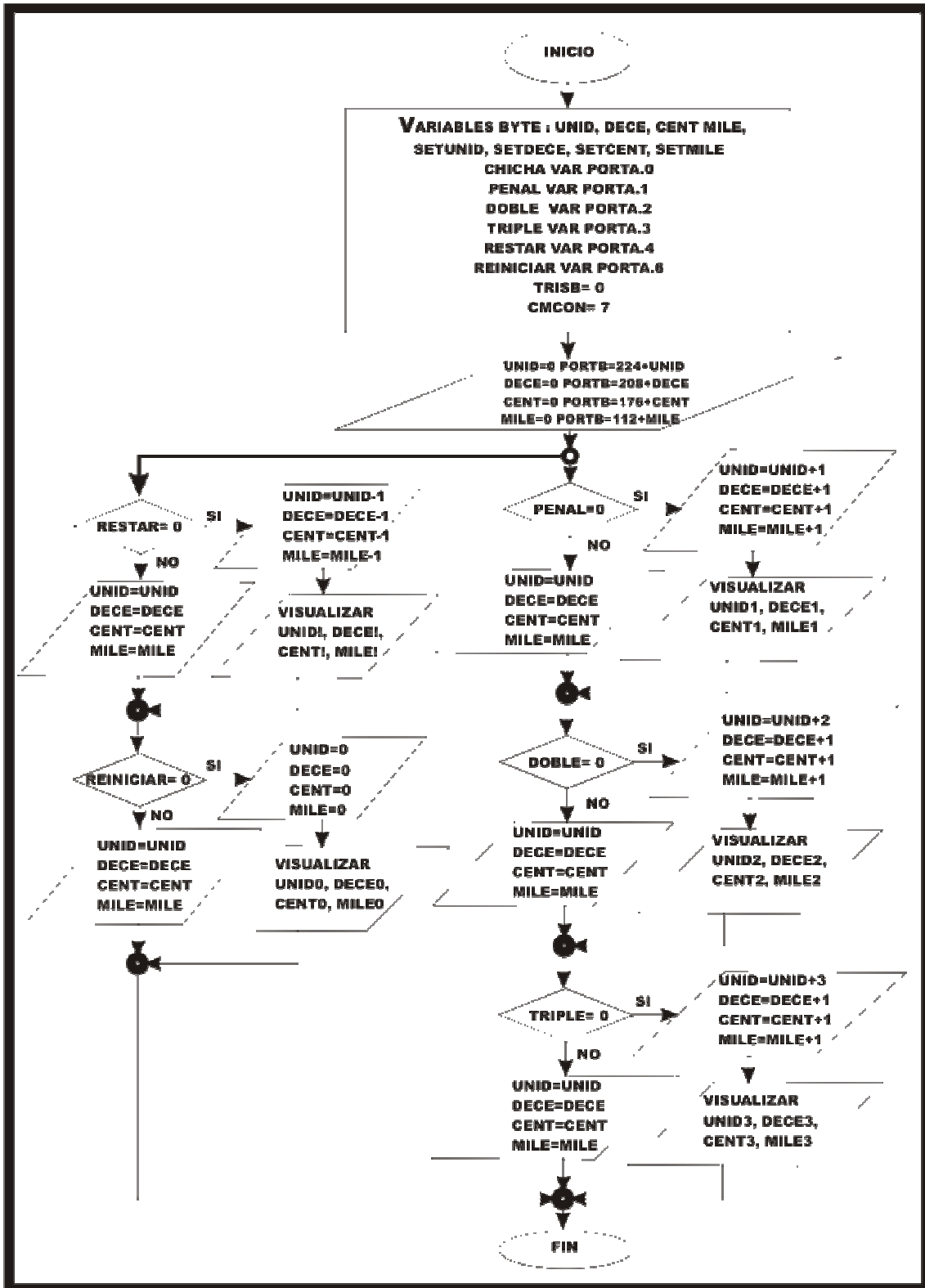


Figura 2.14. Diagrama de Flujo del Contador de aros A/B

2.3.5. EXPLICACIÓN DEL PROGRAMA DEL CONTADOR DE AROS DEL EQUIPO A/B

En este diagrama de flujo de igual manera se empieza definiendo las variables para almacenar los números con (unid, dece, cent, mile), así mismo, se asigna nombres a las líneas del puerto A, en este caso la línea 0 se le asigna chicha y sirve para activar a la chicharra cada vez que se incremente en uno, dos o tres los aros respectivamente; la línea 1 se le asigna el nombre penal y sirve para el incremento en uno, la línea 2 el nombre doble y es el incremento en dos, a la línea 3 el nombre triple y el incremento es en tres. Para el caso de que se requiera reiniciar se utiliza a la línea 6 o en el caso de una equivocación en el incremento de los aros la línea 4, la variable activar que es de 1 bit nos sirve para el antirrobo de las teclas, es decir evitar que cuente más de una vez cuando se mantenga presionado el pulsador; luego se hace a todo el puerto B como salidas con $\text{trisb}=0$ y se convierte todo el puerto A en modo digital para la entrada de datos.

En el desarrollo del programa se hace cero a (unid, dece, cent, mile), para luego proceder con su respectiva activación; después a través de los condicionantes se va a revisar el estado de las variables penal, doble, triple, restar o reiniciar). Cuando $\text{penal}=0$ se incrementa en uno a las unidades, decenas y centenas respectivamente, se enciende la sirena como aviso del incremento y luego se apaga, cuando $\text{doble}=0$ se incrementa en dos unidades al contador se enciende la sirena por un período de tiempo más largo que en el caso de penal, en el caso de $\text{triple}=0$ se incrementa en tres unidades al contador, la sirena se activará por más tiempo que en el doble y luego se apaga. Con $\text{equivocación}=0$ se resta en uno a las unidades, decenas o centenas según corresponda; en cambio cuando $\text{encerrar}=0$ se hace cero a todo el contador y se retorna al inicio del programa para revisar si se cumple una de las condiciones. Cuando no se cumpla ninguna condición el programa permanece revisando el estado de todas las variables.

2.3.6. PROGRAMA DEL CONTADOR DE AROS DEL EQUIPO A/B

; Contador de aros del Equipo A/B

; Definición de variables

unid var byte

dece var byte

cent var byte

mile var byte

setunid var byte

setdece var byte

setcent var byte

setmile var byte

setunid =0 ;# que se requiere el aviso 0130

setdece =3

setcent =1

setmile =0

; Definición De Variables Para Los Terminales Del Puerto A

chicha var porta.0 ;pin de la chicharra

penal var porta.1 ;pulsos para contar

doble var porta.2 ;tecla encerar

triple var porta.3 ;visualizar el valor a comparar

restar var porta.4 ;aro triple

reiniciar var porta.6 ;aro doble

activar var bit ; variable bit para el antirrobote de tecla

trisb=0 ;todo el puerto b es de salida

cmcon=7 ;todo el puerto A en modo digital

 high sirena ;enciende la sirena

 pause 200

; Desarrollo del programa

encerar:

```
unid=0
dece=0
cent=0
mile=0
low sirena      ; apaga la sirena
```

display:

```
portb= 224+unid  ;224 %11100000,activa las unidades
pause 5
portb= 208+dece  ;208 %11010000,activa las decenas
pause 5
portb= 176+cent  ;176 %10110000,activa las centenas
pause 5
portb= 112+mile  ;112 %01110000,activa los miles
pause 5
gosub teclas     ;revisa el estado de las teclas
```

goto display

teclas:

```
if penal      =0 then sumar uno
if restar     =0 then restar uno
if reiniciar  =0 then encerar
if triple     =0 then sumar tres
if doble      =0 then sumar dos
activar=1
```

return

sumar uno:

```
if activar=0 then return ; bandera de tecla penal
activar=0           ; bandera de tecla penal cuando ya ha sido pulsada
unid=unid+1        ; suma 1 a las unidades
high sirena        ; active la sirena
pause 100
low sirena         ; apaga la sirena
if unid<10 then comparar ; si unid<10 hace cero a las unidades
```

```

unid=0
dece=dece+1          ; suma uno a las decenas
if dece<10 then comparar ; si dece<10 hace cero a las decenas
dece=0
cent=cent+1          ; suma uno a las centenas
if cent<10 then comparar ; si cent<10 hace cero a las centenas
cent=0
mile=mile+1          ; suma uno a los miles
if mile<10 then comparar ; si mile<10 hace cero a los miles
mile=0
return

```

restar uno:

```

if activar=0 then return ; bandera de antirrebote de tecla
activar=0
unid=unid -1          ; resta uno a las unidades
high sirena           ; enciende la sirena
pause 100             ; retardo de 100 milisegundos
low sirena            ; apaga la sirena
if unid<10 then comparar ; si unid<10 entonces hace unid=9
unid=9
dece=dece-1           ; resta uno a las decenas
if dece<10 then comparar ; si dece<10 entonces hace dece=9
dece=9
cent=cent-1           ; resta uno a las centenas
if cent<10 then comparar ; si cent<10 entonces hace cent=9
cent=9
mile=mile-1           ; resta uno a los miles
if mile<10 then comparar ; si mile<10 entonces hace mile = 9
mile=9
return

```

sumar tres:

```

if activar=0 then return ; bandera de antirrebote de tecla

```



```

activar=0
unid=unid+3           ; suma tres a las unidades
high sirena           ; enciende la sirena
pause 300              ; retardo de 300 milisegundos
low sirena             ; apaga la sirena
if unid<10 then comparar ; si unid<10 entonces resta 10 a las unidades
unid=unid-10
dece=dece+1           ; suma uno a las decenas
if dece<10 then comparar ; si dece<10 hace cero a las decenas
dece=0
cent=cent+1           ; suma uno a las centenas
if cent<10 then comparar ; si cent<10 hace cero a las centenas
cent=0
mile=mile+1           ; suma uno a los miles
if mile<10 then comparar ; si mile<10 hace cero a los miles
mile=0
return

```

sumar dos:

```

if activar =0 then return ; bandera de antirrebote de tecla
activar =0
unid=unid+2           ; suma dos a las unidades
high sirena           ; enciende a la sirena
pause 200              ; retardo de 200 milisegundos
low sirena             ; apaga la sirena
if unid<10 then comparar ; si unid<10 resta 10 a las unidades
unid=unid-10
dece=dece+1           ; suma uno a las decenas
if dece<10 then comparar ; si dece<10 hace cero a las decenas
dece=0
cent=cent+1           ; suma uno a las centenas
if cent<10 then comparar ; si cent<10 hace cero a las centenas
cent=0
mile=mile+1           ; suma uno a los miles

```

```
if mile<10 then comparar ; si mile <10 hace cero a los miles  
mile=0
```

comparar:

```
if unid!=setunid then return ; compara el número almacenado en unid!  
if dece!=setdece then return ; compara el número almacenado en dece!  
if cent!=setcent then return ; compara el número almacenado en cent!  
if mile!=setmile then return ; compara el número almacenado en mile!  
high sirena ; enciende la sirena  
pause 5000 ; retardo de 5000 milisegundos  
low sirena ; apaga la sirena  
return ; retorna a revisar el estado de las teclas  
return  
end ; fin del programa
```

Nota: Para el caso del contador de aros del equipo A/B, se toma en cuenta lo siguiente; que el incremento de aros normalmente se los realiza de dos en dos a través del pulsador doble, el pulsador triple es en el caso del aro triple, luego el pulsador penalización para el incremento en uno, así mismo el pulsador equivocación para restar uno y finalmente el pulsador de reinicio para volver a cero todo el contador.

2.4. CONSTRUCCIÓN DEL EQUIPO

Para la construcción del equipo se procede principalmente a pasar los circuitos impresos previamente diseñados, a las tarjetas de cobre o más conocidas como baquelitas.

2.4.1. TARJETAS DE CIRCUITO IMPRESO

Primero se imprime a láser en acetatos los circuitos impresos del cronómetro, aquí debido al espacio físico de la tarjeta solo se coloca 3 displays y se tiene que añadir una tarjeta adicional con un solo display, el contador de aros del

equipo A y B; y también se realiza una tarjeta denominada circuito de control la cual posee todos los pulsadores para el control a distancia del marcador electrónico.



Figura 2.15. Tarjetas de circuito impreso

2.4.2. MONTAJE DE LOS ELEMENTOS

Luego se procede a realizar los huecos respectivos en los orificios indicados para cada elemento, dependiendo del diámetro de los pines del elemento se utiliza brocas de 0.5mm hasta de 1.5mm, después se procede a seleccionar todos los elementos que están involucrados en las tarjetas como son los PICs con sus respectivos zócalos, los decodificadores de igual manera, las resistencias, los terminales o conectores, los transistores y lo más importante los displays; para finalmente realizar el montaje de los elementos y soldarlos verificando siempre su posición y valor indicado en el diseño.

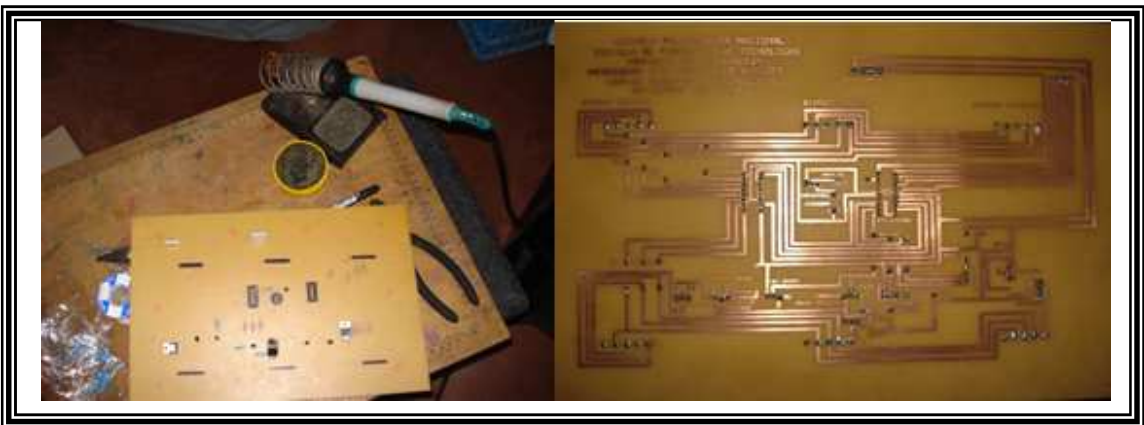


Figura 2.16. Montaje de los elementos

2.4.3. MONTAJE Y CONEXIÓN DE LAS TARJETAS

Aquí se procede a conectar y probar que todas las tarjetas estén funcionando correctamente, para lo cual se utiliza una fuente switching de 500w ATX las que comúnmente se utilizan en los computadores Pentium 4 por su buen rendimiento en voltaje y corriente; dicha fuente posee los voltajes requeridos que son de +5V, tierra y +12V.



Figura 2.17. Montaje y conexión de las tarjetas

2.4.4. ENSAMBLAJE DEL MARCADOR ELECTRÓNICO

Como se puede observar se realiza la conexión de todas las tarjetas, también se debe conectar el circuito de control a través de 15m de cable UTP hacia la tarjeta del cronómetro la cual tiene el control de todos los pulsadores; de tal manera que se realiza el ensamblaje del marcador electrónico de basket.

Ahora bien para poder mejorar la presentación de todo el equipo se realiza el montaje de las tarjetas sobre una caja metálica de las siguientes dimensiones 1.10m x 30cm x 18cm, a la cual se le protege con tela al interior para aislarla y evitar cortocircuitos; al mismo tiempo exteriormente para mejorar la presentación. Por detrás de la caja se coloca la fuente switching, un interruptor de encendido y apagado de la misma; dos terminales RJ-45 hembra. Por la parte de abajo se le instaló la sirena y por delante se coloca un acrílico transparente, sobre el cual se le coloca una pancarta para indicar el contador de aros del equipo A y B, el cronómetro que señala los dos displays para los segundos y los otros dos para los minutos.



Figura 2.18. Ensamblaje del Marcador Electrónico

Para el circuito de control también se tiene que elaborar una caja de madera, con un acrílico en la parte de arriba; por donde están colocados todos los pulsadores y conectores RJ-45 para su interconexión con el marcador electrónico, quedando el equipo terminado de la siguiente manera.



Figura 2.19. Equipo terminado

2.5. PRUEBAS Y AJUSTES

2.5.1. PRUEBAS

En esta parte se realizan diferentes pruebas del equipo terminado, se le coloca dentro de un lugar cerrado y en lugar abierto; en el primer caso funciona correctamente con una luminosidad muy apreciable situándolo a unos 15m de distancia y una altura de 3m, el cronómetro se ajusta al ritmo de un reloj normal y el incremento de aros doble, triple y penalización fue el adecuado sin equivocaciones.

Se le realizan pruebas con el incremento de aros, doble, triple y el de penalización para escuchar la duración del sonido de la sirena, dicho sonido es perfectamente distinguible entre los aros anotados.

2.5.2. AJUSTES

En el segundo caso en un lugar abierto donde ingresan los rayos del sol y la claridad en gran magnitud fue difícil distinguir los números de todo el marcador, es por eso que se le coloca una visera de 30cm de longitud y se le inclinó 10cm hacia adelante para que cree una sombra y pueda observarse el marcador, de esta manera se le pudo divisar a unos 10m de distancia y 2m de altura; es así que funciona de igual manera que lo anteriormente descrito como en un lugar cerrado.

Se le realiza un ajuste en el tiempo de juego del cronómetro para que coordine adecuadamente con el tiempo de un reloj que comúnmente se utiliza, de esta manera el porcentaje de error al culminar el partido es lo más mínimo, es decir en segundos.

2.6. COSTO DEL EQUIPO

CANTIDAD	ELEMENTOS	V. UNITARIO \$	V. TOTAL \$
10	Displays 9*12cm	19.50	195
3	PIC16F628A	3.50	10.50
3	Decodificadores 7447	0.50	1.50
21	R = 100 / 0.5w	0.02	0.42
43	R = 4.7k / 0.5w	0.02	0.86
10	Transistores 2N3904	0.05	0.50
10	Transistores 2N3906	0.05	0.50
10	Transistores TIP127	0.50	5.00
1	Transistor TIP122	0.50	0.50
2	Transistores ECG268	1.50	3.00
1	Sirena 12V / 30W	4.80	4.80
11	Pulsadores	0.10	1.10
2	Interruptores	0.25	0.50
3	Baquelitas 20 x 30cm	3.00	9.00
2	Baquelitas 20 x 10 cm	1.00	2.00
15	Acido férrico	0.40	6.00
1	Fuente switching 500W	20.00	20.00
1	caja metálica 1.10m x 30cm x 17cm	55.00	55.00
6	Zócalos de 18 pines	0.20	1.20
5	Regletas de 40 pines	1.20	6.00
30m	Cable UTP cat. 5e	0.30	9.00
4	Jack RJ-45	0.50	2.00
4	Conectores RJ-45	0.10	0.40
4	Conectores molex 4 pines	0.50	2.00
2	Conectores de 5 pines	0.40	0.80
1	Conector de 10 pines	1.00	1.00
2	Conectores de 8 pines	0.60	1.20
1	Pancarta 1.10m x 30cm	5.00	5.00
	TOTAL		\$ 344.78

Figura 2.20. Costo del Equipo



CAPÍTULO III

3. CONCLUSIONES Y RECOMENDACIONES

3.1. CONCLUSIONES

Para el correcto funcionamiento tanto de los displays y circuitos integrados del marcador electrónico se coloca una fuente switching que suministra +12V para la polarización a los displays y +5V para la polarización de los decodificadores y PICs, dicha fuente se la utiliza por su buen rendimiento ya que posee en +12V una corriente de 20A y en +5V una corriente de 15 A; dependiendo de la carga que se le coloque.

Se utiliza tres PICs 16F628A debido, principalmente a su fácil manejo y programación, también porque físicamente resulta más conveniente distribuir sus pines en cada etapa del circuito y sobre todo porque se necesita del uso de sus dos puertos el A y B, tanto para el cronómetro y los dos equipos el A/B.

Para el manejo de los displays de ánodo común se utiliza el decodificador 7447, el cual tiene sus salidas en nivel bajo y activan correctamente a los segmentos de dicho display, con una corriente de 45mA.

Los displays utilizados son de 9x12 cm, los cuales poseen una correcta luminosidad y funcionan con un voltaje de 12V con su respectiva resistencia de protección, la cual es de 100 ohmios.

En las etapas de los drivers de corriente se utiliza un arreglo de transistores bipolares NPN y PNP, así como también un transistor Darlington encargado de encender el display, este arreglo se lo realiza acoplando los transistores en cascada de tal manera que todos trabajen en corte y saturación; para que finalmente activen al transistor Darlington con el voltaje y corriente adecuados para un correcto encendido del display respectivo.

En el circuito de control se utiliza pulsadores sensibles, que se activan en cualquier dirección que sean presionados, así como también un interruptor con

luz piloto de indicación de funcionamiento del cronómetro, todo esto para brindar un fácil manejo del equipo.

La sirena que indica el incremento de los aros y los tiempos de juego reglamentarios es de 12V/30w, la cual se activa por un tiempo determinado dependiendo del tipo de aro y el tiempo de juego que se haya terminado de jugar.

Se utilizó el software EXPRESS PCB por su fácil manejo, al realizar pistas en forma manual o en el caso de realizarlas en forma automática con su respectivo circuito esquemático, todo esto para el diseño de los circuitos impresos por computadora, estos diseños se los imprime a láser en láminas de acetato, los cuales mediante la transferencia térmica se traslada a las tarjetas de cobre.

Para poder observar y proteger a los 10 displays que conforman el marcador electrónico se coloca una lámina de acrílico transparente en el frente de todos.

Mediante el uso del cableado UTP cat. 5e para la interconexión del marcador con el circuito de control se puede llegar a extender los cables hasta 100m. de distancia sin ningún problema de interferencia, porque las normas de este cableado indican estos parámetros.

3.2. RECOMENDACIONES

Se recomienda que el marcador electrónico sea utilizado en coliseos cerrados para aprovechar al máximo la luminosidad de los displays, en el caso de que sea utilizado en un lugar abierto se sugiere colocarlo en el interior de un cobertor o una cubierta para que se observe de una mejor manera.

Para la utilización del circuito de control se han dispuesto de 11 pulsadores y 1 interruptor, los cuales son de fácil manipulación, se activan en cualquier dirección arriba, abajo, derechas, izquierdas y al centro; por lo que se sugiere que sean activados en forma correcta sin presionarlos demasiado para evitar daños.

Para el caso de que se requiera conectar el equipo a 220 VAC se recomienda utilizar el selector de voltaje que posee la fuente switching, para evitar daños en el equipo.

Si se requiere mejorar la visualización del marcador electrónico en el caso de ser utilizado en lugares abiertos como canchas deportivas, la manera más efectiva es cambiar los displays, por displays formados por diodos leds de alta luminosidad.

El equipo posee 15m. de cable UTP cat. 5e, para la interconexión entre el marcador electrónico y el circuito de control, si se desea extender el cableado se debe colocar empalmes de conectores RJ-45 y a su vez colocar cable UTP de la extensión que se requiera y poncharlos en forma directa a los cables para garantizar un correcto funcionamiento del equipo.

Es recomendable que se utilice agarraderas sujetas correctamente a la pared o algún sitio donde vaya a ser fijado el marcador electrónico, de tal manera que se aseguren con las manijas que fueron colocadas previamente en su superficie; todo esto para obtener una correcta visualización dependiendo a la altura que se ha colocado.

BIBLIOGRAFÍA

¹ Corrales, Santiago. (2005), " *Electrónica práctica con microcontroladores PIC*", primera edición, Ecuador.

² DUQUE, Edison. (1998), " *Curso avanzado de microcontroladores*", Editorial Tecnológica, Pereira.

³ Gonzáles, José. (1996), " *Introducción a los microcontroladores*", MacGrawHill, México.

⁴ Reyes, Carlos. (2004), " *Aprenda rápidamente a programar microcontroladores PIC*", primera edición, Ecuador.

Arquitectura del PIC16F628A

⁵ <http://www.microchip.pic16F628A.pdf.com>

Cableado UTP

⁶ <http://www.monografias.com/trabajos5/ponchado/ponchado.shtml>

Características eléctricas del display LMS-40101BS

⁷ <http://www.ucontrol.com.ar/Articulo20/Display4x7seg/4x7seg.htm>

Características Eléctricas de los transistores Bipolares

⁸ <http://www.nteinc.com/specs/100to199/pdf/nte123ap.pdf>

Características Eléctricas de los transistores Darlington

⁹ <http://www.nteinc.com/specs/200to299/pdf/nte261.pdf>

Características Eléctricas del transistor Bipolar nte268

¹⁰ <http://www.nteinc.com/specs/200to299/pdf/nte268.pdf>

Características de los marcadores de venta

¹¹ <http://www.baybor.com/marcador-deportivo-amateur.html>

Fuente switching

¹² http://www.dbup.com.ar/tutorial_fuentes_conmutadas.htm

Manejo de decodificadores

¹³ <http://docs-europe.origin.electrocomponents.com/webdocs/020c/09007.pdf>

Set de instrucciones del PIC16F628A

¹⁴ <http://www.oshonsoft.com/downloads.html>

Teoría del PIC16F628A

¹⁵ <http://ciecfie.epn.edu.ec/PROYECTOS/pdfs/Explorador1.pdf>

Tipos de display de 9x12cm

¹⁶ <http://www.cypsela.es/especiales/pdf195/display.pdf>

Transistores Bipolares, Darlington NPN y PNP

¹⁷ http://www.unicrom.com/Tut_transistor_bipolar.asp



ANEXOS



ANEXO A

CARACTERÍSTICAS DE LOS MARCADORES DE VENTA

En este parte se muestra los distintos marcadores electrónicos que existen a la venta en la actualidad, la tecnología utilizada, sus características tanto de visibilidad como de uso; en lugares abiertos como cerrados; sean estos coliseos o canchas deportivas.

Este es un marcador electrónico para basket de la categoría amateur el cual posee el cronómetro, marcador local, visitante, número de faltas e indicación del período de juego.



Marcador Electrónico¹¹

CARACTERÍSTICAS FUNCIONALES
Equipado de fábrica con una reja metálica que protege la parte frontal del marcador de cualquier impacto de pelota o balón
Marcador integrado en una caja extra plana especialmente diseñada para poder ser instalada fácilmente en cualquier pared, soporte o columna
Este modelo solamente se fábrica en versión para interiores y equipa dígitos de diferentes colores
Cronómetro programable en sentido creciente o decreciente (0:00 - 99:59)
Reloj horario con función alarma incorporada
Tanteo local y visitante de 0 a 199 puntos
Número del período en juego de 1 a 4 períodos
Número de sets ganados por cada equipo
Señalización del equipo en poder del servicio

CARACTERÍSTICAS TÉCNICAS
Dimensiones del marcador: 120 x 70 x 8 cm
Caja exterior metálica INOXVER a la que se le aplica la pintura especial antirreflectante de color negro mate que facilita el contraste necesario para que los dígitos tengan una excelente visibilidad
Sujeción del marcador a pared mediante una barra metálica transversal ubicada en su parte trasera-superior (anclajes, tornillería y tacos incluidos)
Tipo construcción electrónica: 100% estado sólido
Dígitos formados por puntos de leds de diferentes colores
Dígitos cronómetro de partido y puntos señalización número de Período en juego formados por leds de color rojo
Dígitos tanteo y faltas acumuladas equipo Visitante formados por leds de color verde
Leds de alta luminosidad de 10 mm de diámetro
Altura dígitos: 20 y 25cm
Visibilidad: 100 mts
Señal sonora: Bocina 90db y 100Hz

Precios De Marcadores Deportivos ¹¹

Serie	Modelo	Altura dígitos	Visibilidad	Dimensiones	Precio sin IVA
Amateur	Sobremesa Interior	12 cm	50 m	60 x 39 x 25 cm	999 euros
Amateur	Sobremesa Exterior	12 cm	50 m	60 x 39 x 25 cm	1.400 euros
Amateur	Mini-Basket	12 cm	50 m	150 x 70 x 5 cm	2.900 euros
Serie	Modelo	Altura dígitos	Visibilidad	Dimensiones	Precio sin IVA
Profesional	Portátil	22 cm	90 m	142 x 35 x 20 cm	1.900 euros
Profesional	Sénior	27 cm	130 m	200 x 100 x 8 cm	2.500 euros
Profesional	Basket	22 y 27 cm	130 m	200 x 150 x 8 cm	3.600 euros

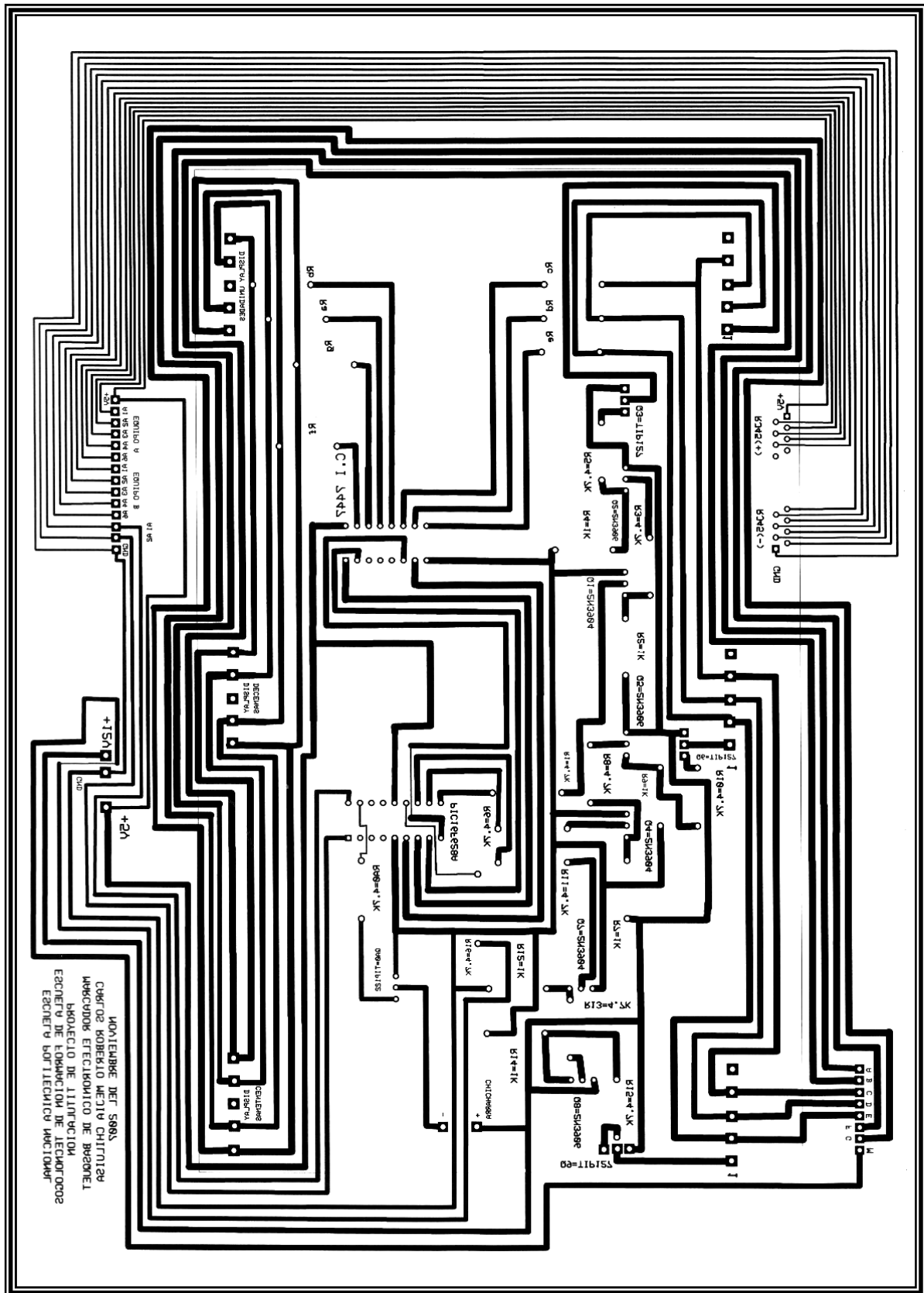
¹¹ <http://www.baybor.com/marcador-deportivo-amateur.html>



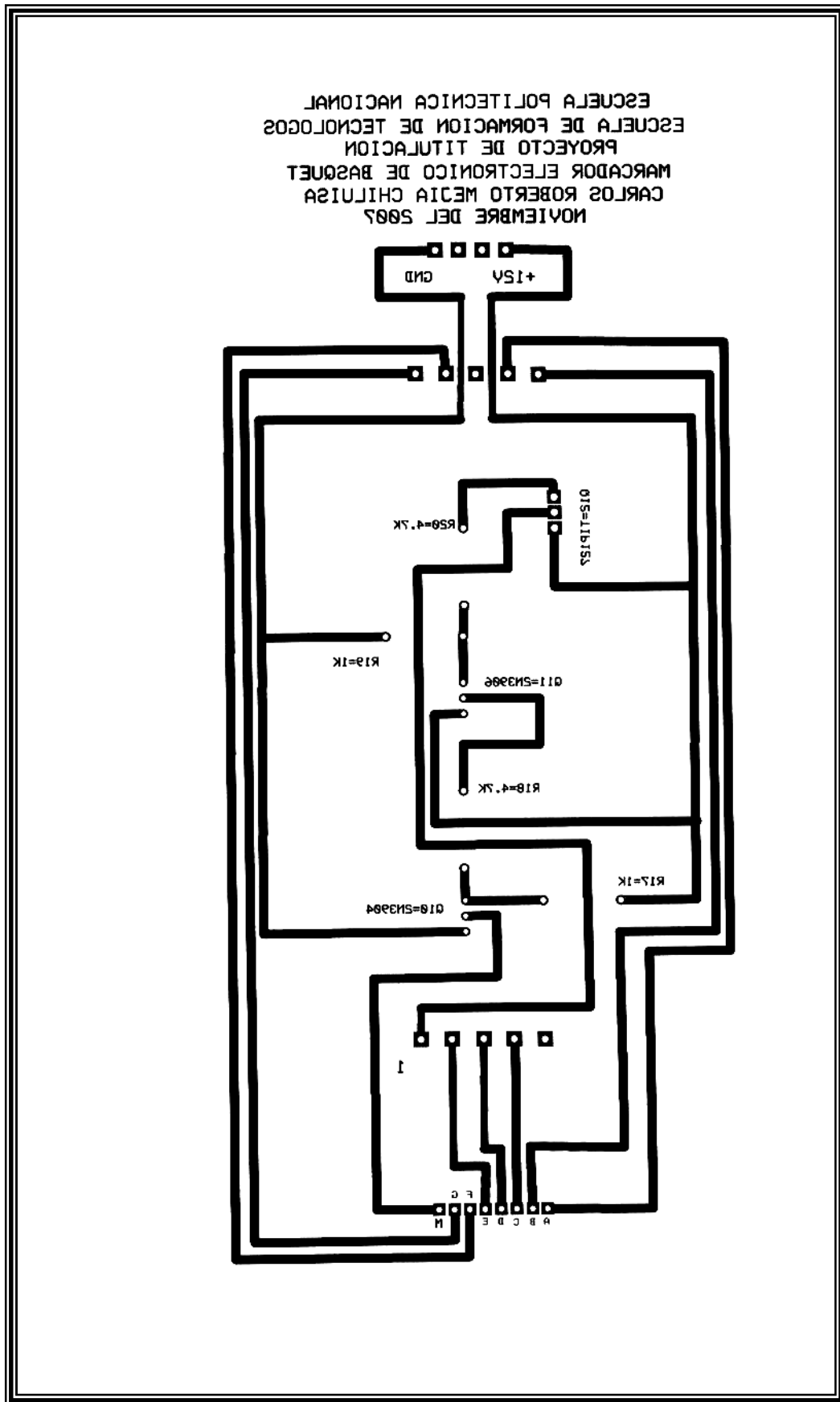
ANEXO B

CIRCUITOS IMPRESOS DEL PROYECTO

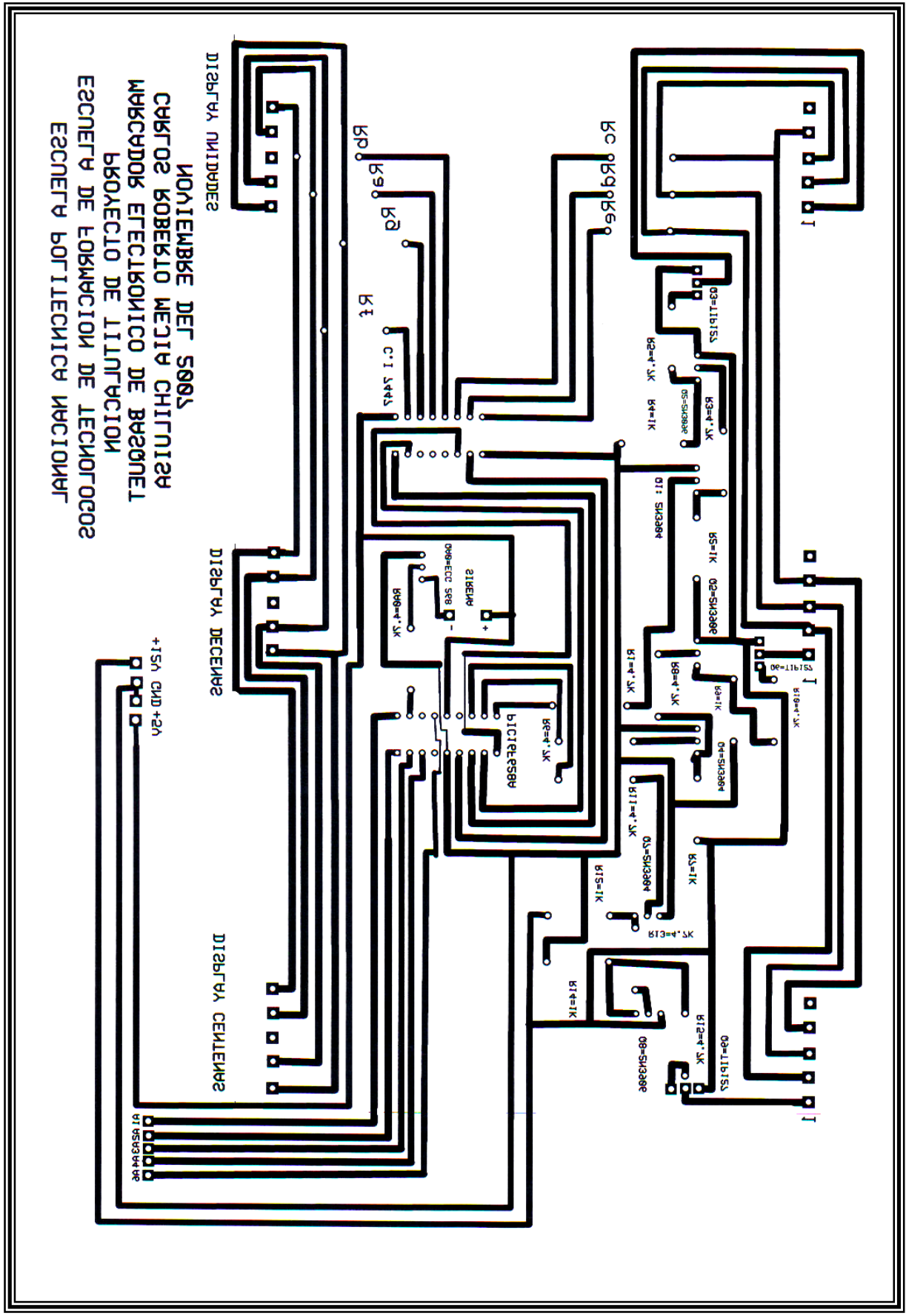
CIRCUITO IMPRESO DEL CRONÓMETRO



CIRCUITO IMPRESO DEL CUARTO DISPLAY DEL CRONÓMETRO



CONTADOR DE AROS DEL EQUIPO A/B



NOVIEMBRE DEL 2002
 CARLOS ROBERTO MEJIA CHILUISA
 PROYECTO DE TITULACION
 ESCUELA DE FORMACION DE TECNOLOGOS
 ESCUELA POLITECNICA NACIONAL

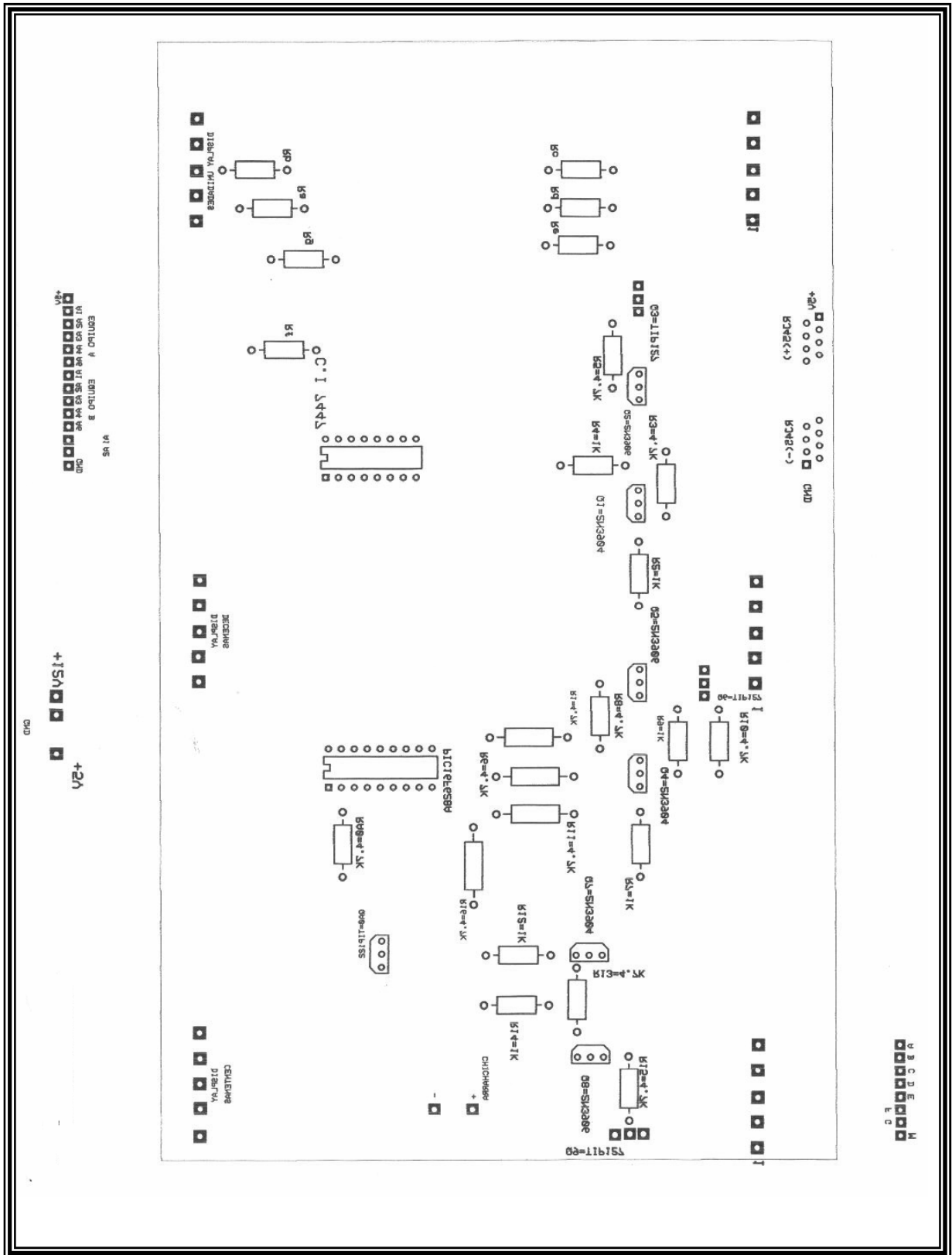
DISPLA Y UNIDADES

DISPLA Y DECENAS

DISPLA Y CENTENAS

+15V GND +2V

DISTRIBUCIÓN DE ELEMENTOS DEL CRONÓMETRO





ANEXO C

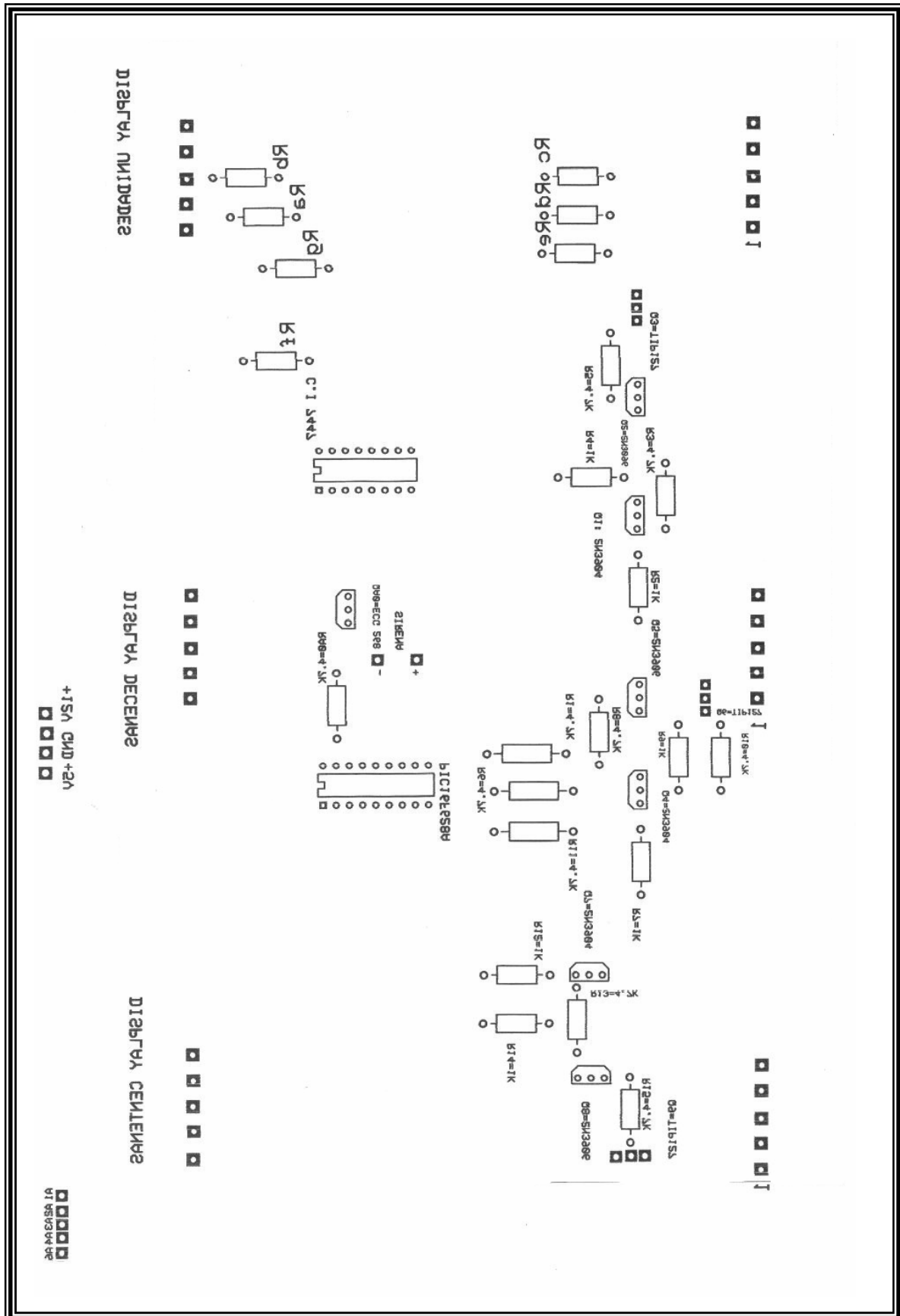
MANUAL DEL EQUIPO

Las partes principales que posee el cronómetro que son susceptibles a dañarse son:

- El PIC16F628A, en el caso de que necesite ser programado o reemplazado, cuando no realice la función como el conteo de minutos, segundos o reinicio del tiempo.
- El decodificador 7447 cuando reciba el código binario del PIC y no active a ningún segmento de los displays, debe ser reemplazado por su respectivo reemplazo u original.
- E el caso de que no encienda algún display se debe verificar que este correctamente polarizado con +12V en su respectivo transistor Darlington de activación para esto se tiene a los transistores Q3, Q6, Q9, Q12 que son el TIP127, el cual en el último de los casos debe ser reemplazado por uno de similares características de corriente y voltaje.
- Cuando los segmentos no se enciendan con una luminosidad adecuada se debe verificar la resistencia correspondiente al segmento averiado para lo cual se tiene las resistencias Ra, Rb, Rc, Rd, Re, Rf, Rg que son de 100 ohmios.
- Se debe verificar que los voltajes de alimentación de +5V, +12V y tierra sean los adecuados desde la fuente switching de alimentación hasta el conector respectivo del cronómetro, la continuidad en los cables o a su vez la correcta medida de voltaje de corriente continua, todo esto para evitar cambiar elementos innecesariamente.
- Se debe verificar los conectores de los displays, es decir, realizar una limpieza con algún líquido limpiador de contactos, ya que por el uso, la humedad o calidad del material se suelen dañar y producen fallas en sus contactos.

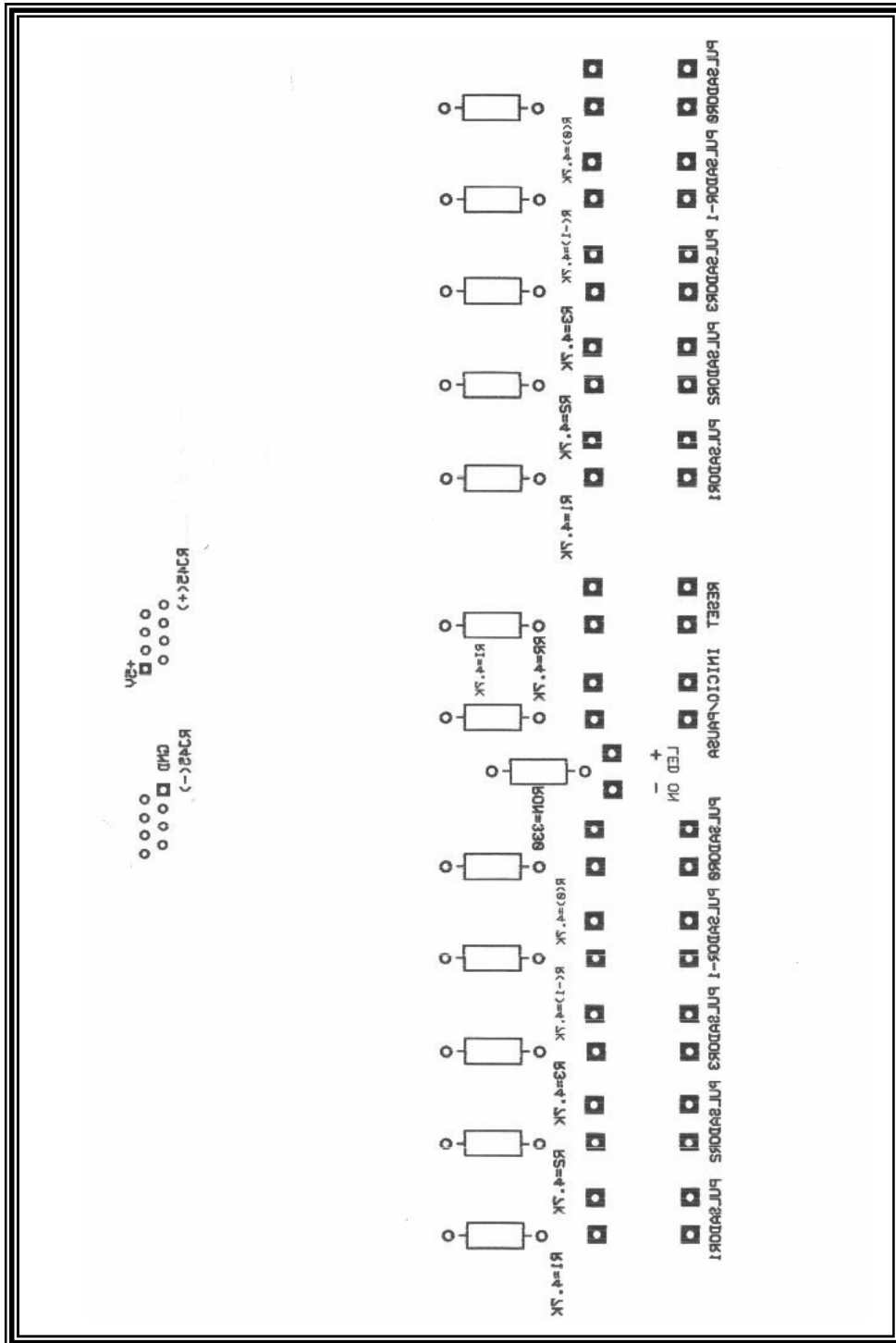
DISTRIBUCIÓN DE ELEMENTOS DEL CONTADOR DE AROS DEL EQUIPO

A/B



De igual manera en el contador de aros del equipo A/B existen partes que son susceptibles a dañarse, al igual que en el cronómetro el contador de aros posee las mismas partes y elementos distribuidos de igual forma, por lo que los daños son similares.

DISTRIBUCIÓN DE ELEMENTOS DEL CIRCUITO DE CONTROL



- En el circuito de control existen 2 conectores RJ-45, los cuales son los encargados de establecer la conexión con el cronómetro que también posee 2 conectores similares, estos conectores se suelen dañar por una mala manipulación o al insertar el conector RJ-45 macho de una manera inadecuada, por lo que se les debe cambiar por unos similares y están ubicados como RJ-45(+) y RJ-45(-) en la distribución de los elementos.

- Los pulsadores suelen dañarse por una manipulación excesiva o inadecuada, para lo cual se les debe cambiar por unos similares que se activen en cualquier posición o que sean fácilmente manipulables por el usuario, por lo que se encuentran identificados desde el pulsador (1, 2, 3,-1,0), tanto para el equipo A como el B, en el caso del cronómetro tiene un interruptor que se le ubica como inicio/pausa y el pulsador de reinicio como reset.



ANEXO D

MANUAL DEL USUARIO

Para la utilización del equipo se necesita conocer las partes principales de las cuales está conformado, entre las que se tiene el panel principal que es el de los displays, el cableado, el circuito de control y la fuente de alimentación.

Introducción

- **Displays:** El panel principal es el de los diez displays, los cuales están distribuidos en tres para el Equipo A, cuatro para el cronómetro (dos para los minutos, dos para los segundos) y tres para el Equipo B; estos son visibles hasta 25m. en lugares cerrados. En la parte trasera este panel principal de displays posee la fuente de alimentación que se conecta a 110VAC, también tiene dos conectores RJ-45 hembra para conectarse al circuito de control a través del cableado UTP.

- **Circuito de Control:** El circuito de control posee 11 pulsadores para el incremento de aros de los dos equipos, 1 interruptor para el inicio o pausa del cronómetro, 1 luz piloto que indica el funcionamiento del cronómetro y además dos conectores RJ-45 hembra para la interconexión con el panel principal de displays.

- **Cableado UTP:** El cableado UTP de interconexión es de la cat. 5e, tiene aproximadamente 2x 15m de largo, posee conectores RJ-45 ponchados en ambas puntas directamente, es decir, son cables directos para la transmisión de los datos de los pulsadores hacia el panel principal de displays.

- **Fuente de alimentación:** La fuente de alimentación es una fuente switching ATX de 500w, que suministra al circuito +12V para los displays y +5v para los circuitos integrados y PICs; la misma que puede ser conectada a 110VAC o 220VAC, según sea seleccionado el voltaje con el selector que posee.

Instalación

La instalación del equipo se la debe realizar en un coliseo o cancha deportiva, el panel de displays posee dos manijas que pueden ser sujetas a cualquier superficie con sus respectivos ganchos a la altura deseada para una correcta visualización; ya que el peso aproximado del panel es de 12kg. lo que facilita su manipulación. Aquí también está ubicada la fuente switching la cual debe ir conectada a través de un cable de alimentación a 110VAC, los dos terminales RJ-45 hembra debe ir conectados según la polaridad que esta señalada.

El circuito de control debe ser apoyado sobre una mesa sea de madera o metálica para su uso, los pulsadores son de fácil accionamiento y el interruptor también, la luz piloto que indica el funcionamiento del cronómetro se ve claramente en la oscuridad o claridad.

El par de cables UTP de 15m. deben ser correctamente extendidos dependiendo de la distancia a la que se vaya a colocar el equipo, los terminales RJ-45 conectados con la polaridad indicada tanto en el panel de displays como en el circuito de control.

Funcionamiento

El equipo entra en funcionamiento cuando se pulsa el interruptor de inicio del cronómetro, se enciende la luz piloto para indicar el constante incremento del tiempo de juego, entonces la persona encargada del circuito de control que sería el juez suplente o vocal del partido de basket, incrementará los aros tanto del equipo A o del equipo B con los diez pulsadores que posee el panel de control, cinco para el un equipo que servirán para incrementar aros de uno, dos o tres según sea el caso, un pulsador para restar un aro y otro para el reinicio del marcador.

En el caso de que se desee reiniciar el cronómetro o parar el partido por algún inconveniente, se debe pulsar el interruptor hacia donde diga pausa y reiniciar o a su vez continuar con el tiempo de juego.

Cada vez que se incrementa en uno, dos o tres los aros la sirena se activará por los tiempos establecidos; también cuando se llega a los períodos de juego y descanso reglamentarios la sirena se activará por más tiempo

MANUAL DE MANTENIMIENTO

En esta parte se debe tener en cuenta que todo el equipo ha sido construido con elementos electrónicos, cables o piezas que ya sea por su uso o mala utilización se podrían llegar a dañar; por lo que se debe saber como proceder a realizar su respectivo mantenimiento lo cual incluye el cambio de elementos o reparación de sus partes o piezas. Tal es el caso de que se dañen los PICs 16F628A, en este caso se debe reemplazarlos por unos similares grabarles el programa adecuado sea el del cronómetro o el contador de aros y volverlos a colocar en los zócalos que fueron soldados en las tarjetas respectivamente.

Cuando no funcione los decodificadores de BCD a 7 segmentos 7447, igualmente se debe reemplazar por unos similares y colocarlos en la posición adecuada según el zócalo del cual fue sustraído.

En el caso de que no encienda alguno de los displays del cronómetro o el contador de aros se debe verificar principalmente que existan los +12V en el ánodo común, antes de proceder a reemplazarlo, en el caso de hacerlo se debe colocarlo en la posición adecuada según la guía que poseen sus respectivos zócalos. A su vez sino se enciende algún display se debe revisar la etapa del driver de corriente que esta conformada por los transistores Darlington y bipolares, se los debe probar uno a uno; en el caso de encontrar alguno en mal estado se lo debe desoldar y reemplazarlo por otros de similares características.

En el cableado UTP se puede producir daños en los conectores RJ-45, para esto se debe cortar el conector defectuoso, colocar otro en la misma posición y poncharlo directamente para su buen funcionamiento.

En el circuito de control se pueden llegar a dañar los pulsadores por su uso, para esto se los debe reemplazar por otros similares, lo importante es que sean fácilmente manipulables.

Para realizar un correcto mantenimiento a continuación se mostrará una tabla con fallas y posibles soluciones que se puede efectuar en el equipo para su correcto funcionamiento.

SÍNTOMA	POSIBLE CAUSA	CORRECCIÓN
No enciende el equipo.	<ul style="list-style-type: none"> ➤ Cable de alimentación en mal estado. ➤ Interruptor de encendido defectuoso. ➤ Fuente en mal estado 	<ul style="list-style-type: none"> ➤ Cambiar el cable. ➤ Cambiar por un interruptor similar. ➤ Cambiar fusible o reemplazarla.
No inicia o no se pausa el cronómetro.	<ul style="list-style-type: none"> ➤ Interruptor de inicio desconectado o defectuoso. 	<ul style="list-style-type: none"> ➤ Reemplazar interruptor o verificar cables de conexión.
No enciende un display del equipo A.	<ul style="list-style-type: none"> ➤ Verificar +12V al ánodo común. ➤ Transistores en mal estado. 	<ul style="list-style-type: none"> ➤ Cambiar cableado ➤ Reemplazar transistor
No funciona el circuito de control.	<ul style="list-style-type: none"> ➤ Cableado UTP en mal estado. ➤ Conectores RJ-45 defectuosos. 	<ul style="list-style-type: none"> ➤ Cambiar cableado o verificar avería. ➤ Cambiar conectores.
No se activa pulsador de aro de dos.	<ul style="list-style-type: none"> ➤ Pulsador dañado 	<ul style="list-style-type: none"> ➤ Cambiar pulsador por uno similar.

Localización de fallas

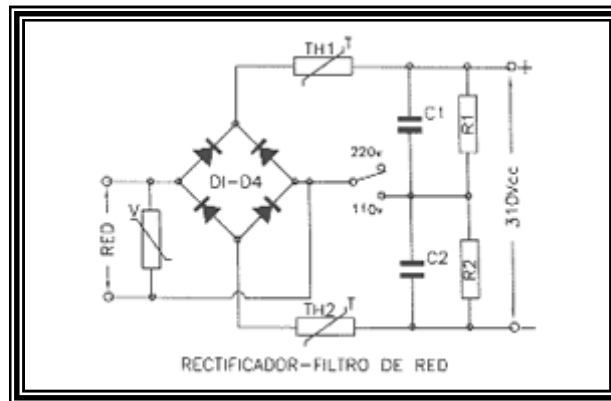


ANEXO E

FUENTE SWITCHING

Filtro de entrada ¹²

Las fuentes conmutadas son convertidores de voltaje, por lo que la red debe ser previamente rectificadas y filtradas con una amplitud de rizado. La mayoría de las fuentes utilizan el circuito de la Figura 3.7 para operar desde 90 a 132 VAC o de 180 a 260 VAC según sea la posición del conmutador.



Fuente switching ¹²

En la posición de abierto se configura como rectificador de onda completa, por lo que se obtiene aproximadamente 310 Vcc desde la red de 220 VAC. En la posición de cerrado el circuito funciona como rectificador doblador de tensión, ahí se obtiene también 310 Vcc a partir de 110 VAC.

Para evitar sobrecalentamientos los condensadores electrolíticos (C1 y C2) deben ser de bajo ESR (baja resistencia interna) y de la tensión adecuada. Es conveniente conectar en paralelo con estos otros condensadores tipo MKP para mejor desacoplo de alta frecuencia de conmutación. Los rectificadores deben soportar una tensión inversa de 600v

Pico de arranque

Al arrancar una fuente conmutada, la impedancia que se presenta en la red es muy baja al encontrarse con los condensadores descargados, sin una resistencia en serie adicional la corriente inicial sería excesivamente alta. En la Figura 3.7, TH1 y TH2 son resistencias NTC (coeficiente negativo de temperatura), que limitan esta corriente a un valor aceptable. Las fuentes de

media y gran potencia disponen de circuitos activos con resistencia limitadora que se cortocircuita por medio de relés o de conmutadores estáticos cuando ya están los condensadores cargados. En el caso de las fuentes de AMV se utiliza un transistor MOS-FET de potencia.

Protección contra transitorios

Además del filtrado de ruidos reinyectados a la red que incorporan las fuentes conmutadas, se utiliza un varistor conectado a la entrada para proteger contra picos de tensión generados por la conmutación en circuitos inductivos de las proximidades o por tormentas eléctricas.

Características de la fuente switching

Para el marcador electrónico se utiliza una fuente ATX de 500W, la cual posee voltajes de +12V, +5V, -12V, -5V, con sus correspondientes valores de corriente que son de 24A, 22A y 0.8A, los cuales a su vez dependen de la carga a la cual se conecten para obtener un mejor rendimiento.



Características ¹²

¹² http://www.dbup.com.ar/tutorial_fuentes_conmutadas.htm



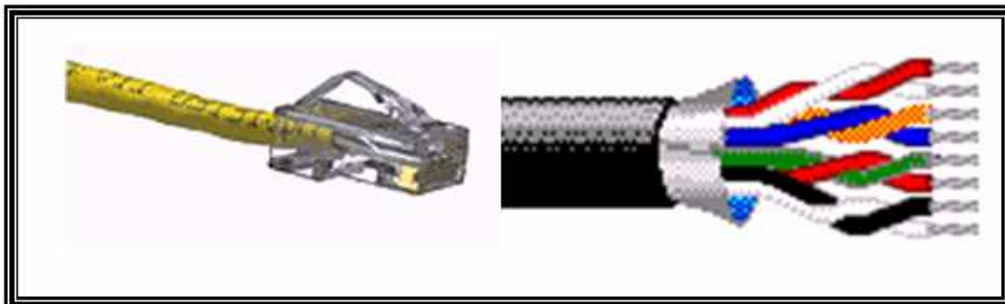
ANEXO F

CABLEADO UTP

UTP son las siglas de (Unshielded Twisted Pair). Es un cable de pares trenzados y sin recubrimiento metálico externo, de modo que es sensible a las interferencias; sin embargo, al estar trenzado compensa las inducciones electromagnéticas producidas por las líneas del mismo cable. Es importante guardar la numeración de los pares, ya que de lo contrario el efecto del trenzado no será eficaz, ya que disminuye la capacidad de transmisión. Es un cable barato, flexible y sencillo de instalar. La impedancia de un cable UTP es de 100 ohmios.

Cable STP: STP son las siglas de (Shielded Twisted Pair). Este cable es semejante al UTP pero posee un recubrimiento metálico para evitar las interferencias externas. Por tanto, es un cable más protegido, pero menos flexible que el primero, el sistema de trenzado es idéntico al del cable UTP. La resistencia de un cable STP es de 150 ohmios.

Estos cables de pares tienen aplicación en muchos campos. El cable de cuatro pares es utilizado como la forma de cableado general en muchas empresas, como conductores para la transmisión telefónica de voz, transporte de datos, etc.



Estructura de un cable UTP ⁶

En los cables de pares hay que distinguir dos clasificaciones:

Las Categorías: Cada categoría especifica unas características eléctricas para el cable: atenuación, capacidad de la línea e impedancia.

Las Clases: Cada clase especifica las distancias permitidas, el ancho de banda conseguido y las aplicaciones para las que es útil en función de estas características.

CLASES	Clase A	Clase B	Clase C	Clase D
Ancho de banda	100 kHz	1 MHz	20 MHz	100 MHz
En categoría 3	2 km	500 m	100 m	no existe
En categoría 4	3 km	600 m	150 m	no existe
En categoría 5	3 km	700 m	160 m	100 m

Características de longitudes posibles y anchos de banda ⁶

El UTP de categoría 5 es barato y fácil de instalar, se está incrementando su utilización en las instalaciones de redes de área local con topología en estrella, mediante el uso de conmutadores y concentradores. Las aplicaciones típicas de la categoría 3 son transmisiones de datos hasta 10 Mbps (por ejemplo, la especificación 10 baseT); para la categoría 4, 16 Mbps, y para la categoría 5 (por ejemplo, la especificación 100 BaseT), 100 Mbps. En concreto, este cable UTP de categoría 5 viene especificado por las características de la figura 3.11 (especificaciones TSB-36) referidas a un cable estándar de 100 metros de longitud.

Velocidad de transmisión de datos	Nivel de atenuación
4 Mbps	13 dB
10 Mbps	20 dB
16 Mbps	25 dB
100 Mbps	67 dB

Nivel de atenuación según la velocidad de transmisión ⁶

⁶ <http://www.monografias.com/trabajos5/ponchado/ponchado.shtml>